
La revue des
SHARPENTIERS
SHARP
RAM CARD 16
N° 14 - BIMESTRIEL - SEPTEMBRE 85 - 30 F.
LA REVUE DES UTILISATEURS DE MICRO-ORDINATEURS ET POCKET-COMPUTERS
SHARP



SOMMAIRE N° 14

La journée EDUCATIQUE 85 a démontré, s'il en était encore besoin, l'énorme potentiel d'avenir que représente l'ordinateur de poche dans la vie quotidienne du professeur et de l'élève. Sa portabilité, son faible coût, ses possibilités de calculs scientifiques souvent supérieures à celles des ordinateurs de table, font du « PORTEPUCE » un outil pédagogique de base au même titre que le tableau noir ou le cahier de cours. L'utilisation rationnelle de ce nouvel outil crée un besoin croissant d'informations et de mise à jour des connaissances acquises. Elle implique également une collaboration étroite entre les différents partenaires, parties prenantes de cette nouvelle pédagogie.

Au Club des Sharpentiers, ce type de coopération est maintenant entré dans une phase très active. A l'image des nombreuses pages ENSEIGNEMENT de ce bulletin, nous nous efforçons de présenter toutes intiatives, expériences ou réalisations créées autour de nos produits micro-informatiques, dans des domaines aussi variés que possible.

Cette rubrique est essentiellement écrite par des enseignants, pour des enseignants ; c'est pourquoi, comptant beaucoup sur votre active participation nous avons mis au point, sur le modèle du critérium sharpentiers, le CRITERIUM ENSEIGNEMENT qui permettra à tout auteur d'être abonné gratuitement pour une nouvelle année au Club et de gagner, éventuellement, l'un des derniers produits SHARP. Nous serons également très attentifs à tous vos conseils, suggestions et propositons ; ils seront une aide essentielle dans notre volonté de créer un véritable partenariat actif.

L'arrivée des « POCKETS » ouvre une nouvelle page de l'histoire de l'enseignement. Une autre page, celle des vacances, appartient, malheureusement, déjà à l'histoire. Alors... Bonne rentrée.

S. BIZOIRRE

CRITERIUM SHARPENTIERS

Les règles en sont très simples mais peuvent rapporter gros aux SHARPENTIERS courageux :

1. Tout Sharpentier dont l'article, le programme ou l'astuce aura été publié dans le bulletin SHARPENTIER sera d'office inscrit au club, gratuitement, pour une nouvelle année.
2. A la parution de chaque bulletin, une main innocente tirera au sort l'un de ces auteurs qui gagnera ainsi l'un des plus récents produits SHARP.

Le tirage au sort du bulletin n° 13 a désigné M. A. OZANNE pour ses nombreuses réalisations sur PC-1500 et PC-1401, il gagne ainsi un PC-1421. Pour le n° 14 il y a un PC-1450 à gagner.
A vos plumes.

ONT PARTICIPE A CE NUMERO

REDACTEUR EN CHEF
S. BIZOIRRE
REDACTEURS
L. BURELLER
M. GIRONDOT
J.-F. VIGNAUD
SECRETAIRE DE REDACTION
G. HAYET

P. ABRIVARD, S. ARNAUD, M. BALLAND, E. BERNARD, E. BEAUREPERE, P. COSMIDES, P. DEHLINGER, M. FOJUD, M. GIRBAL, B. KOKANOSKI, D. MAGNIN, C. MARTIN, P. MATSIS, J. MAZET, J. MILLET, C. MUNCH, A. OZANNE, D. PERRET, X. PIERSON, C. POULAIN, Y. RAVELOMANATSOA, H. SZCZEPANSKI, F. UBALDI, S. VISENNE, G. WOLFHUGEL

MAQUETTE ET FABRICATION IN-QUARTO. 19, RUE F.LEMAITRE 75020 PARIS. TEL. 636.73.04

Club des Sharpentiers 151/153, avenue Jean-Jaurès 93307 AUBERVILLIERS CEDEX Tél : 834.93.44

LES NOUVEAUTÉS

PA 3260

La dernière née des machines à écrire portables électroniques de chez SHARP à ROUE D'IMPRESSION

Grâce à sa roue d'écriture de 98 caractères, la **PA 3260**, l'une des plus petites portables du marché actuel, offre toutes les qualités d'impression d'une machine à écrire de bureau.
Un écran cristaux liquides de 16 caractères permet le contrôle du texte frappé avant son impression.
La correction automatique sur papier est illimitée.

Ses fonctions spécifiques :
- Alinéa automatique
- Centrage automatique
- Soulignement automatique
- Caractères renforcés
- Tabulation simple et décimale
- Justification à la marge droite

La **PA 3260** est une machine à écrire mais aussi... à calculer.

Cette machine fonctionne sur le secteur et par batterie rechargeable CdNi d'une autonomie de 20 000 caractères environ.

LES NOUVEAUTÉS

LES NOUVEAUTÉS



Un autre avantage : cette petite machine se connecte directement sur un ordinateur par interface RS 232 C.
Plus petite qu'une attaché-case (367 x 307 x 59 mm) et très légère (3,2 Kg), cette machine vous accompagnera lors de vos déplacements.

La PA 3260 est idéale pour les mémoires, les rapports et les courriers traditionnels.



Prenez 1/3 de PC 1401, ajoutez 2/3 de PC 1350, mélangez délicatement et laissez mijoter le tout quelques jours dans les bureaux d'études de SHARP et vous obtenez... un PC 1450.
Sa taille est celle du PC 1350, son design, celui du PC 1401. De ces deux produits il tire les avantages. Du 1401, il prend le mode calcul et les nombreuses fonctions scientifiques et statistiques. Du 1350, il empreinte la RS 232 TTL et devient le premier micro-ordinateur de poche scientifique communiquant. Il utilise également le système de mémoire par cartes, offrant un choix plus étendu de cartes : 2, 4, 8 ou 16 Ko. C'est du côté du concept mémoire que nous vient la véritable nouveauté. Il n'y a plus de mémoire vive dans la machine ; tous les programmes sont logés sur cartes, c'est pourquoi tout PC 1450 est livré d'office avec une carte mémoire 4 Ko.
Le PC 1450, c'est le véritable outil évolutif du scientifique.

TAILLE : 72 x 182 x 16 mm
POIDS : 175 g
AFFICHEUR : 1 ligne de 16 caractères
CAPACITE MEMOIRE : 2, 4, 8 ou 16 Ko.
MEMOIRE UTILISATEUR : 3070 octets avec carte 4 Ko. ((carte − 1) x 1024-2)

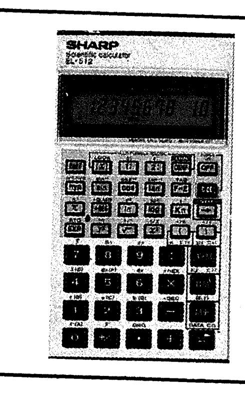


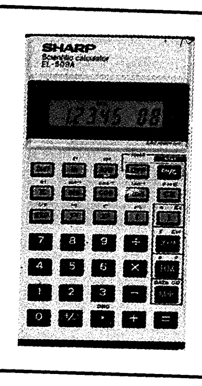
LES NOUVEAUTÉS

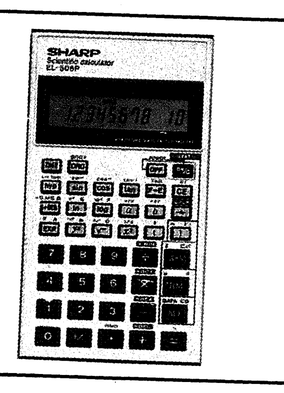
LES NOUVEAUTÉS



LES CALCULATRICES SCIENTIFIQUES







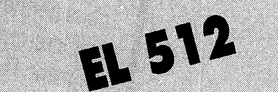
Ce ne sont pas réellement des nouveautés, mais ces machines n'ont jamais été présentées complètement dans notre bulletin. Or, en cette période de rentrée des classes, il est utile que leurs principales caractéristiques vous soient rappelées. Les 4 machines présentées sont proposées dans le cadre de la campagne enseignement et représentent un éventail assez complet de l'ensemble des possibilités que l'on peut attendre d'une calculatrice scientifique.

EL 509 A

Calculatrice scientifique et statistique équipée de 32 fonctions et d'un afficheur 8 chiffres permettant l'affichage scientifique sur 5+2 chiffres. Machine scientifique de base, elle ne connait pas les calculs hyperboliques et les conversions polaires.

EL 531

Caractéristiques très voisines de la EL 509 A, possède 2 fonctions scientifiques supplémentaires.



EL 512

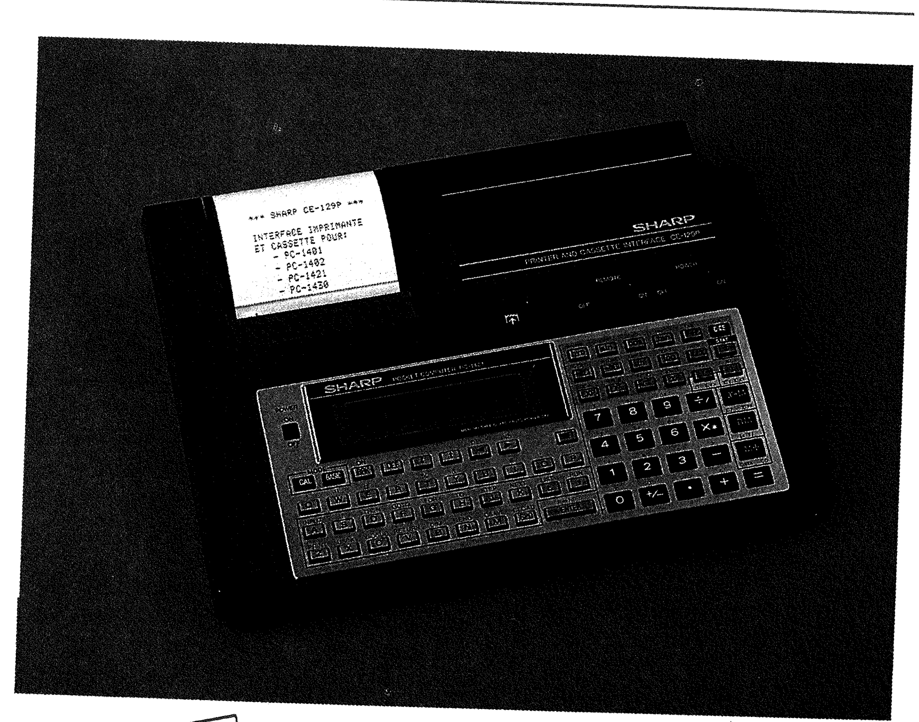
Calculatrice scientifique et statistique programmable sur 128 pas, en programmation linéaire. Elle possède 9 mémoires, 61 fonctions et le même affichage que la EL 506 P. Dans un volume très réduit, c'est la plus puissante des calculatrices de poche.

EL 506 P

Calculatrice scientifique et statistique possédant 56 fonctions scientifiques, un afficheur 10 chiffres permettant l'affichage scientifique sur 8 + 2 chiffres. C'est également l'outil idéal pour le programmeur car elle possède les fonctions de changement de bases (binaire, octal, héxadécimal et décimal).

LES NOUVEAUTÉS

CE 129 P



Fini les fils qui trainent partout, les magnétophones délicats. Le berceau imprimante-interface cassette pour les PC scientifiques est arrivé. Conçu pour s'adapter parfaitement aux PC 1401, 1402, 1421 et 1430, ce berceau formera, avec votre PC, un ensemble compact et facilement transportable. Le CE 129 P est équipé d'une imprimante thermique 24 caractères/ligne sur papier de 54mm et d'une interface cassette (micro, écouteur, télécommande). Le PC qui s'y insère prend une position légèrement inclinée pour faciliter l'utilisation de son clavier.

Les affaires club sont essentiellement constituées de matériels, accessoires ou périphériques qui ne sont pas (ou plus) commercialisés par le réseau de revendeurs SHARP. Comme vous pouvez le constater, leurs prix sont très attractifs. Une précision, toutefois, seules, les « Affaires club » sont commercialisées par le club, à l'exclusion de tous autres matériels normalement distribués par les points de ventes SHARP.



SHARP MZ 720

+ BASIC SHARP
+ SUPER BASIC
+ EDIFORTH
+ 10 LOGICIELS DE JEUX

1 290 F

Oui, vous avez bien lu ! Nos accords avec la Société S.B.M. nous permettent de vous proposer ce prix jamais vu pour le micro-ordinateur MZ 700 et les 13 logiciels qui l'accompagnent. De nombreux développements et programmes ont été réalisés au Club et présentés dans nos différents bulletins. D'autres sont en cours. Sur simple demande de votre part, nous vous ferons parvenir une documentation complète concernant l'ensemble des éléments de cette offre.

SFD

Unité de disquettes 5 POUCES 1/4 connectable directement sur MZ 700. Capacité de stockage : 320 Koctets formatés. Livrée avec DISC BASIC, programmes utilitaires et manuel.



PC 1500

CE 155	Module mémoire 8 Ko.	250 F
CE 153	Table à digitaliser. Extension de 140 touches sensitives au clavier du PC 1500	550 F
QCN*	Connecteur mâle 60 broches permettant l'accès au bus du microprocesseur	215 F

PC 1251

CE 12 A	Micro-cassette et manuel (en Anglais) traitant de 19 programmes orientés STATISTIQUES	120 F
CE 12 B	Micro-cassette et manuel traitant de 20 programmes orientés CALCULS ELECTRONIQUES	120 F
CE 12 C	Micro-cassette et manuel traitant de 20 programmes orientés MATHEMATIQUES	120 F

PC 1211

CE 122	Interface imprimante	490 F

MZ 80 A

AEU	Panier d'extentions pour cartes disquettes et imprimantes	500 F
AFI	Cartes interface disquettes	600 F
AMD	Master disquette	150 F
ADO2	FDOS avec notice très détaillé pour programmer en assembleur sous disquettes	600 F

MZ 80B

EU	Panier d'extentions pour cartes graphiques imprimantes et disquettes	600 F

MZ 80 K

T41	Langage PASCAL (cassette + notice)	200 F
TW	Assembleur (cassette et notice)	200 F

TOUS MZ 80

MZ 80 FDA1	Lecteur de disquettes 5 pouces 1/4 pour MZ 80 K, A ou B	2 200 F
MZ 80 FDA2	Double lecteur de disquettes 5 p.	3 200 F

BON DE COMMANDE

Je passe commande de :

REF.	QUANT.	DESIGNATION	P.U TTC	P.TOTAL TTC
			TOTAL TTC	

NOM .... PRENOM ....
ADRESSE ....

Ci-joint mon règlement de .... F, par chèque bancaire ou C.C.P. établi à l'ordre du SHARP BUROTYPE MACHINES*. Il représente le montant total TTC de ma commande. Je prends bonne note qu'au cas ou ma commande ne pourrait être honorée dans la limite des stocks disponibles, je serai intégralement remboursé du montant des articles non livrés.

DATE .... SIGNATURE

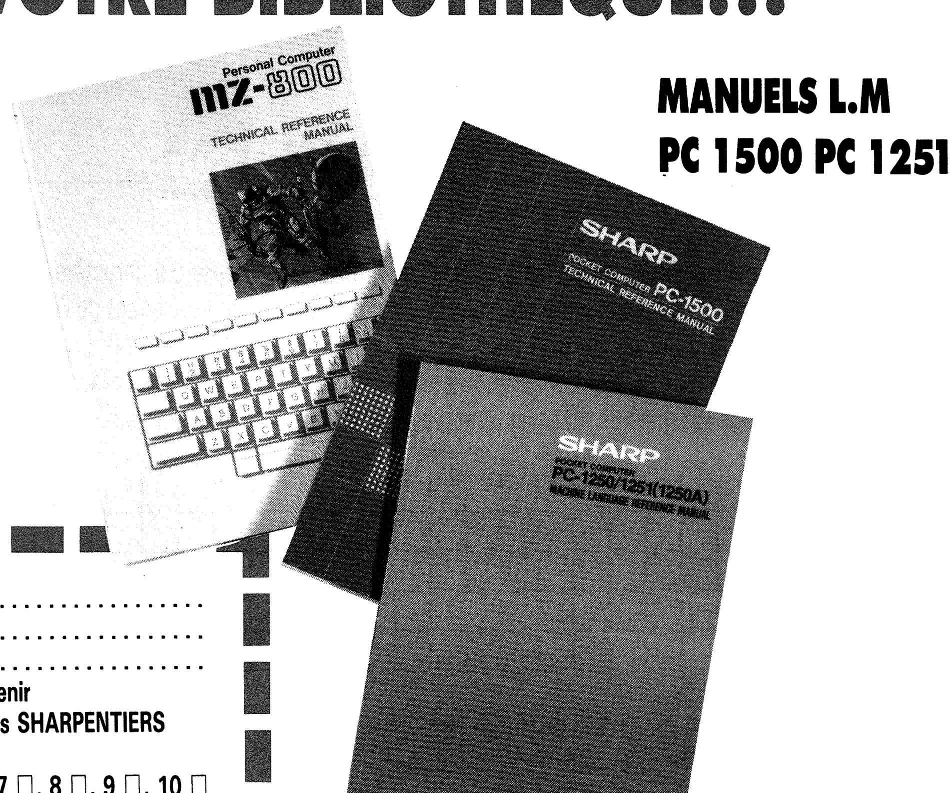
* Pour le connecteur QCN, règlement à l'ordre du « Club des Sharpentiers ».

DANS VOTRE BIBLIOTHEQUE...

TECHNICAL REFERENCE MANUAL MZ 800

Nouvelle exclusivité du Club, ce manuel aborde en détails la gestion de tous les périphériques internes et externes au MZ : Clavier, disquettes, Q.D., mémoire vidéo, couleurs, joysticks, générateur de sons, imprimantes, magnétophone, etc. Il fournit un listing commenté des 2 moniteurs en ROM, une description des circuits internes et une liste détaillée des commandes du Z80. Bref... un ouvrage de référence pour qui veut utiliser à fond les nombreuses ressources du MZ 800.

MANUELS L.M PC 1500 PC 1251



M. ....
ADRESSE ....
....
vous prie de bien vouloir lui faire parvenir
- ☐ Recueil(s) des n° 1 à 5 des bulletins SHARPENTIERS au prix de 110 F. (port compris)
- ☐ Bulletin(s) SHARPENTIER n° 6 ☐, 7 ☐, 8 ☐, 9 ☐, 10 ☐ au prix de 20 F chacun
- ☐ Bulletin(s) SHARPENTIER n° 11 ☐, 12 ☐, 13 ☐ au prix de 25 F chacun
- ☐ Manuel(s) Langage Machine PC 1500 au prix de 180 F l'un
- ☐ Manuel(s) Langage Machine PC 1251 au prix de 180 F l'un
- ☐ Manuel de référence MZ800 au prix de 240 F l'un

Signature

Ci-joint un chèque de.......... francs.
Date

Ces trois manuels, en Anglais, très bien documentés, sont des ouvrages de référence et n'abordent pas l'initiation au langage machine. Ils sont disponibles directement au Club, le mercredi après-midi (exclusivement), au prix de 170 F (MZ800 : 230 F) l'un ; ou par correspondance au prix de 180 F (MZ800 : 240 F) (port compris).

Le manuel LM PC 1251 s'applique également à la série PC 14XX, aux PC 1260, et au PC 1350.



MEFIANCE...

Un important vol de pockets PC 1261 et d'interfaces CE 125 a été commis en région parisienne. Soyez très vigilent s'il vous est proposé l'un de ces produits, surtout s'il ne peut pas vous être fourni de facture ou si le numéro de série a été falsifié ou effacé. Au besoin, contactez le Club.

LE COURRIER

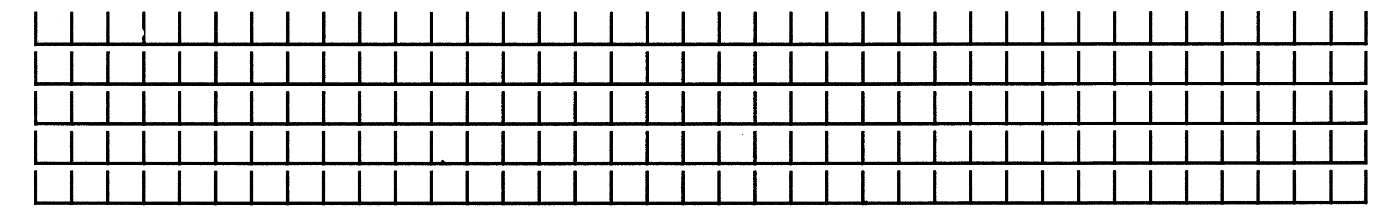
Un gros problème dans la gestion de notre courrier a provoqué de nombreuses perturbations et un retard important dans les réponses aux différentes demandes concernant les MZ 700 et 80K. Ce retard devrait, à la parution de ce bulletin, être totalement comblé. Cependant, si votre demande, datant de plus d'un mois était, à ce jour, restée sans réponse, veuillez nous le faire savoir, nous ferons en sorte de vous donner immédiatement satisfaction. Toutes nos excuses pour ce contretemps qui ne devrait plus se reproduire à l'avenir.

SICOB BOUTIQUE 85

Le SICOB BOUTIQUE déménage. Il se tiendra, cette année, au palais des congrès de la porte MAILLOT. Nous y serons, bien sûr, présents au stand T93, niveau 1. Venez nous voir nombreux, vous découvrirez les derniers logiciels implantés sur PC et MZ ainsi que les dernières réalisations et créations du Club. SHARP sera également présent au SICOB professionnel qui conserve sa traditionnelle implantation au palais du CNIT.
SICOB CNIT LA DEFENSE du 18/9 au 27/09/85
SICOB BOUTIQUE Palais des CONGRES du 23/9 au 28/09/85
L'entrée au SICOB BOUTIQUE est gratuite.



Exclusivement réservées aux particuliers pour la vente ou la recherche de matériel d'occasion, votre annonce devra nous parvenir 1 mois avant la parution du bulletin.
Elle devra impérativement utiliser l'original ou une photocopie de la grille ci-dessous (5 lignes de 40 caractères), à raison d'une case par caractère ou espace. Elle mentionnera, en cas de vente, le type exact de l'appareil, la date d'achat, le prix de vente ainsi que vos coordonnées complètes.
Le Club se réserve le droit de refuser toute annonce ne répondant pas aux critères ci-dessus.



Ci-joint le texte de mon annonce à paraitre dans le prochain bulletin, accompagné de son règlement (80 F) à l'ordre du club des SHARPENTIERS.

Vends PC 1 500 83 8K + CE-150 + MICRO K7 + LM + prgs. TBE. Le tout 3 500 F.
Alain RAPAPORT - La Boudinière - 42330 St-Galmier
(77) 54.13.52

Vds PC 1211 (1981) + revues du club : 900 F.
S. MONFORIES SP 7489 BONN — ALLEMAGNE.
Recherche progs. astrologie sur PC 1350 avec carte 16K + CE 516 P.

Vends MZ 80 A 83. 2 800 F avec manuels et logiciels. Peut être vendu avec panier extensions interface disquettes et disquettes.
Monsieur DIDELOT
(1) 871.20.43 ou contacter le club.

Informatique et enseignement technique

Il y a bien longtemps (une quinzaine d'années) que certains professeurs de disciplines techniques s'intéressent de près aux moyens de calculs informatiques pariculièrement adaptés à leurs préoccupations. En effet, contrairement à d'autres qui parfois se contentent d'obtenir une expression littérale, le technicien recherche avant tout des valeurs numériques. Celles-ci expriment les caractéristiques et les dimensions de la construction ou de la machine qu'il conçoit ou réalise.

L'apparition des calculettes scientifiques bon marché a marqué une première étape il y a quinze ans ce fut la fin de la règle à calculs. Mais à l'époque, les machines programmables (consoles ou poquettes) étaient très chères, de faible capacité, et assez difficiles à programmer en langage machine spécialisé. Si le professeur pouvait montrer aux élèves les possibilités des moyens modernes de calcul, il était rare que ceux-ci aient l'occasion d'appuyer eux-mêmes sur les touches du clavier.

Tout a basculé il y a trois ou quatre ans. C'est à ce moment que le parc des machines de poche transportées par les étudiants des grandes classes dans leur serviette a commencé à comporter une majorité de programmables. Laisser inemployée une telle potentialité aurait été irrationnel, même s'il fallait aussi proposer des algorithmes particuliers aux élèves démunis de ce type de matériel.

Et c'est immédiatement que les effets se firent sentir. Le premier : les abaques, les tables de valeurs, autrefois d'un usage courant, perdirent soudain beaucoup de leur importance. Pourquoi s'embarasser d'une pile de paperasses alors que dix lignes de basic donnent le résultat plus rapidement, avec une précision bien meilleure ? Le maître écrit et explique le programme au tableau (ou mieux, passe le listing au rétroprojecteur) ; les étudiants l'adaptent à leur matériel personnel. Quelques exemples bien choisis permettent de débusquer les bogues... Belle économie sur les photocopies !

Deuxième conséquence : toute une cargaison de formules approchées, concoctées au cours des âges pour faciliter les calculs manuels, n'ont plus de raison d'être. La poquette donne aussi rapidement et facilement le résultat à l'aide de l'expression exacte, sans que l'on ait à se soucier des limites de validité de l'approximation.

Troisième utilisation en classe avec les élèves : le tracé des courbes point par point, tâche autrefois fastidieuse à cause des calculs répétitifs. Le programme une fois établi et essayé, chacun peut tracer sa courbe avec toute la précision désirée. Par exemple, lors de l'étude du système bielle-manivelle, l'élève peut établir le diagramme des vitesses du piston. Mais pas de n'importe quel piston ! du piston du moteur de son propre véhicule, dont il suffit d'entrer au clavier la course et la longeur de bielle. Et pour ce genre de problèmes, l'usage ultérieur d'une informatique de plus fortes dimensions permet de visualiser sur un écran ou de tracer sur une table les courbes obtenues. On peut aussi se demander quel est l'avenir de techniques de calcul comme la statique graphique si souvent utilisés par le passé, pour étudier les poutres en flexion. En fait, dans les conditons actuelles, une disparition complète semble exclue, mais par contre le champ d'application risque fort d'être dorénavant limité à des questions simples.

Mais pourquoi en rester là ? Nous utilisons l'informatique de poche avec les élèves, nous pouvons aussi l'utiliser contre eux ! Car dans ces matières où les résultats numériques sont si importants, les travaux remis avaient trop souvent une bien fâcheuse tendance à se ressembler... Le problème-type de résistance des matériaux comporte en départ le calcul d'une dimension minimale, d'où l'on déduit une valeur « ronde », qui sert de base pour la vérification des déformations. L'énoncé contiendra donc des données sous forme littérale ; l'ordinateur, grâce à un premier programme, choisit, imprime et mémorise, pour chaque élève, des valeurs numériques personnalisées prises au hasard, mais néanmoins de façon judicieuse. Plus aucune copie n'est semblable. Et la correction ? Pas plus ennuyeuse qu'autrefois : un deuxième programme, de correction cette fois-ci, rappelle à lui les données correspondant à la copie, demande les choix effectués, affiche ou imprime les résultats numériques cherchés. Il reste alors à comparer, et à chercher éventuellement l'endroit où l'élève s'est trompé.

En école d'ingénieurs, le principe est le même pour les notes de calculs d'avant-projet, mais la taille et le but du travail sont différents. Trente à quarante questions au total sons rassemblées en cinq ou six groupes ; et il ne s'agit pas tant de « mettre une note » que d'entraîner des étudiants qui ont déjà fait leurs preuves. Chaque élève-ingénieur résoud son problème personnel pas-à-pas et demande au professeur de vérifier ses résultats partiels au fur et à mesure. Celà évite qu'une étourderie en cours de route ne conduise à un ensemble farfelu. Evidemment, il faut un programme bien structuré. Car il serait fastidieux, pour vérifier le troisième groupe de questions, d'avoir à refaire tous les calculs déjà précédemment testés...

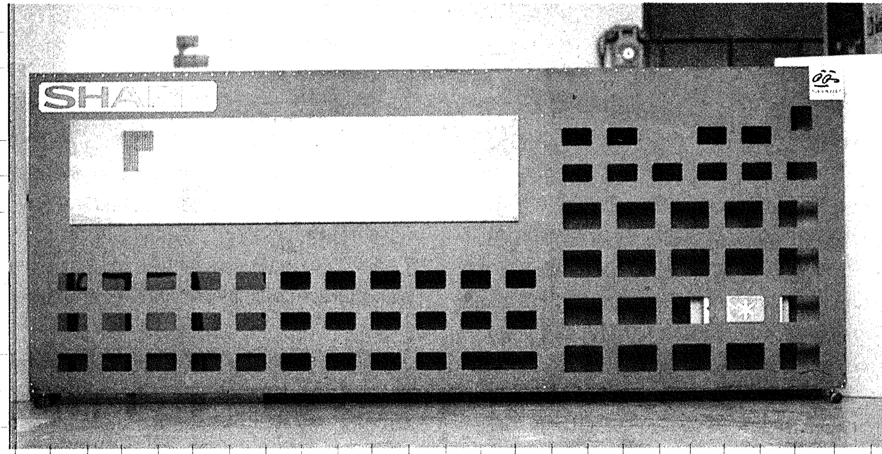
Les ordinateurs de poche peuvent aussi servir à informatiser le travail administratif du professeur. Par exemple, en début d'année scolaire, celui-ci entre dans le désordre les noms de ses nouveaux élèves. Le système les classes par ordre alphabétique, imprime la liste, la sauvegarde sur bande magnétique. A la fin du trimestre, un deuxième programme rappelle cette liste depuis le magnétocassette, demande pour chaque élève le nombre de notes obtenues, la valeur de chacune de ces notes. Une fois cette opération réalisée, l'opérateur a le temps d'aller faire un petit tour ; mais lorsqu'il reviendra, il trouvera imprimé le classement de la classe par ordre de mérite, avec la moyenne de chacun et en prime la moyenne de la classe (éventuellement même l'écart-type).

Nous ne prétendons pas que ces méthodes soient dès aujourd'hui généralisées, mais elles sont en tous cas déjà largement diffusées. Tout ceci apprtient au monde actuel et n'a rien à voir avec la science-fiction. Sans consoles ni écrans, sans stages lourds ni légers, sans propagande échevelée, les professeurs de nos lycées techniques et de nos écoles d'ingénieurs se sont bel et bien appropriés ce merveilleux instrument de travail qu'est l'informatique de poche. Et ils attendent de pied ferme tous les progrès techniques qui nous sont promis pour demain dans ce domaine.

DANIEL MAGNIN

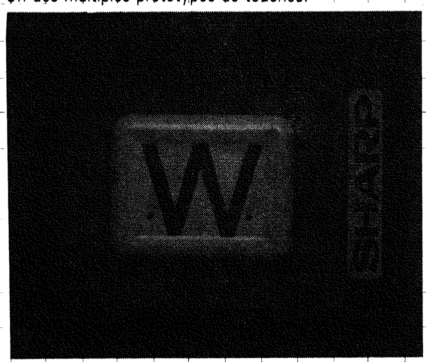
Un Giga-Micro

Une idée « surréaliste », un esprit d'entreprise à toute épreuve, et une bonne dose de génie, tels sont les ingrédients qui auront suffi à Monsieur NEYTCHEFF, professeur au lycée technique de Melun, pour mettre en œuvre son « colossal » projet : la réalisation, à l'échelle 15 d'un ordinateur de poche en parfait état de marche!

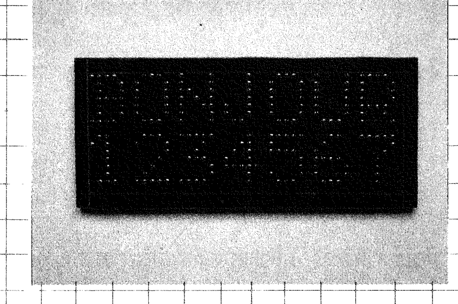


Imposant... pour un pocket.

Un des multiples prototypes de touches.



Les tests d'afficheurs.



Colossal n'est pas un vain mot ; imaginez plutôt un PC 1350 de 2.4 mètres de long, muni de toutes ses touches d'environ 15 cm chacune, disposant d'un affichage composé d'un réseau de 4800 diodes Led ; l'ensemble relié à son petit frère d'origine, intégré dans la « boîte » par une bonne centaine de mètres de fils. Imaginez la greffe d'un moteur de Solex dans un autobus et vous aurez une mince idée des difficultés rencontrées. Ces difficultées, Monsieur NEYTCHEFF les a balayées une à une en faisant participer tous les corps de métiers représentés dans son lycée. Professeurs et élèves des industries des métaux et des plastiques, électroniciens, électriciens, menuisiers, tous ont collaboré à l'élaboration de ce projet.

Seule, une opération n'a pu être réalisée au sein de l'établissement : les soudures au laser des connections au vrai PC 1350, effectuées au microscope par une société spécialisée. Le Club et la Société S.B.M. se sont, bien sûr, associés pour une grand part à cette réalisation ; l'Education Nationale, également puisque ce projet a fait l'objet d'un P.A.E. (Projet Action Educative) présenté par l'établissement.

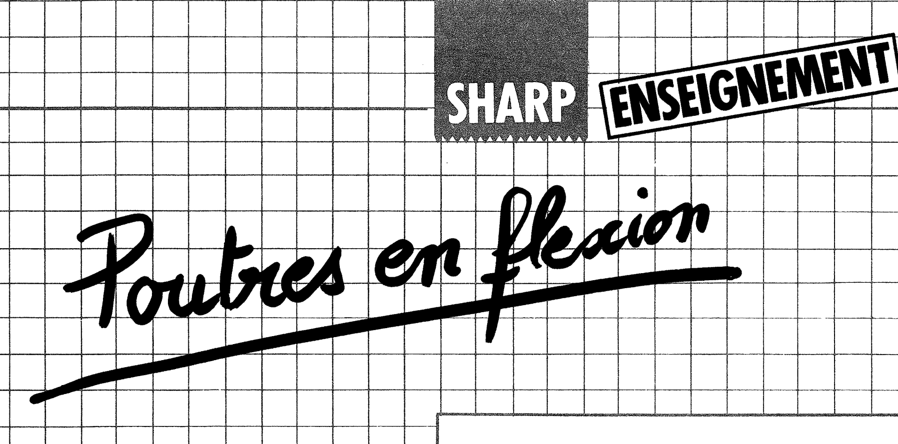
Les premiers essais sont actuellement en cours, s'ils sont concluants, vous pourrez venir vous-même programmer ce mastodonte sur le stand SHARPENTIERS du SICOB BOUTIQUE où il effectuera sa première sortie. Dans notre prochain numéro, nous vous présenterons un second P.A.E., d'un style très différent, réalisé autour d'un SHARP MZ 730 : La simulation d'une entreprise dans une classe de couture.

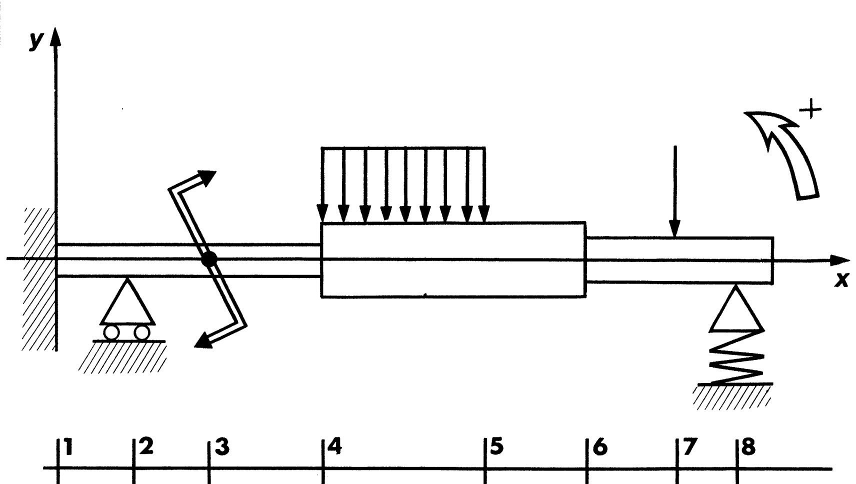
Si vous envisagez également, au sein de votre établissement, un projet alliant l'informatique à toute forme de pédagogie, n'hésitez pas à nous contacter, nous mettrons volontiers en œuvre nos moyens techniques et notre expérience au service de vos idées.

S.B.



« On est prié d'apporter ses gants de boxe »





• Toute poutre peut-être schématisée par des éléments déformables. Dans le shéma ci-dessus, il y a 7 éléments compris entre 8 nœuds.

• Chaque nœud correspond à un *appui* avec l'extérieur ; un *point d'application* d'effort. [Force, moment ou les deux]; une *variation* de forme ; un début ou fin d'application des charges réparties.

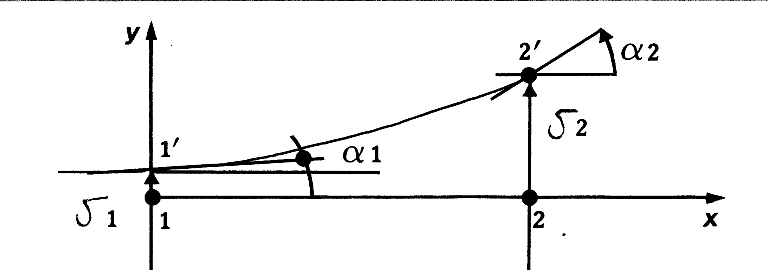
• Convention d'axes

Repères orthonormé direct x,y,z.

Sur la figure

- Moment en 3 est négatif
- Charge répartie (4 → 5) est négative
- charge concentrée en 7 est négative.

I. RELATION [FORCES], [DEPLACEMENTS]



L'élément 1-2, après déformation va prendre la position 1'-2'.

La distance 1-1' sur y est la flèche $\delta 1$.

La rotation $\alpha 1$ par rapport à x est la rotation de la section 1.

De même 2-2' sur y est la flèche $\delta 2$.

$\alpha 2$ est la rotation de la section 2

T1 = Force exercée par le nœud (1) sur (1-2)

M1 = Moment exercé par le nœud (1) sur (1 – 2)

T2 = Force exercée par le nœud (2) sur (1 – 2)

M2 = Moment exercé par le nœud (2) sur (1 – 2)

Les relations forces/déplacements sont de la forme

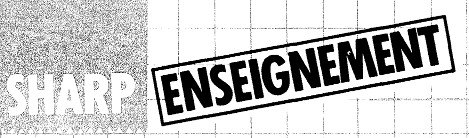
$$\begin{bmatrix} T1 \\ M1 \\ T2 \\ M2 \end{bmatrix} = \frac{EIgz}{L3} \begin{bmatrix} 12 & 6L & -12 & 6L \\ 6L & 4L^2 & -6L & 2L^2 \\ -12 & -6L & 12 & -6L \\ 6L & 2L & -6L & 4L^2 \end{bmatrix} \begin{bmatrix} V1 \\ \alpha 1 \\ V1 \\ \alpha 2 \end{bmatrix}$$

E : Module d'élasticité longitudinale de l'élément considéré.

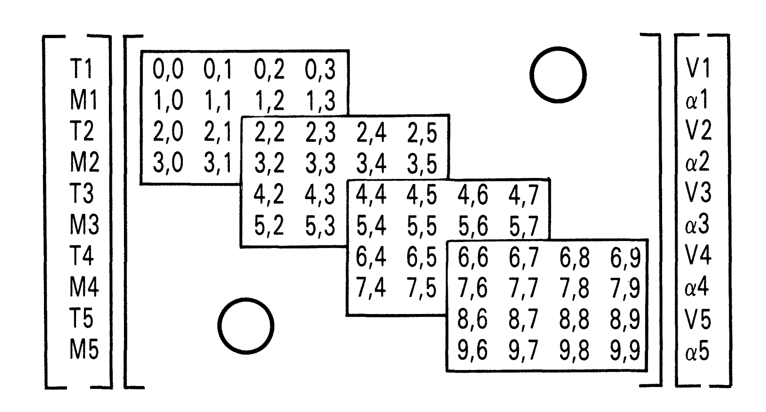
L : Longueur de l'élément considéré.

Igz : Moment quadratique de l'élément.

Le nœud numéro 1 intervient une seule fois. Les nœuds suivants devront intervenir dans l'élément qu'ils terminent et celui qu'ils commencent. Le dernier n'intervient qu'une seule fois.



La matrice *finale* sera donc :



Cas d'une autre poutre, avec 4 barres, donc 5 nœuds.
Ensuite, il faut résoudre le système.

II. TYPES DE LIAISON, ENTREE DES DONNEES

Sur le schéma numéro 1, le nœud numéro 1 représente un encadrement. Dans ce cas V et α sont nuls. On trouvera alors T et M.
Le nœud numéro 2 est une liaison appui simple. Dans ce cas V seul est nul. On trouvera T et α.
Le nœud numéro 8 est sur appui élastique. On introduira le cœfficient de raideur k.
T = kV. On trouvera T, V et α.
Les autres nœuds sont libres ; en 3 on introduit un moment seul ; en 7 une force seule ; en 4,5,6 ni moment ni force. On trouvera α et V.

Résumé : Classification
Type 0 : Libre : introduire Force, moment.
Type 1 : Appui simple : introduire moment.
Type 2 : Elastique : introduire K, moment
Type 3 : Encastrement.

III. DONNEES RELATIVES A CHAQUE ELEMENT

Chaque type de nœud étant désigné, il faut alors introduire les données relatives à chaque élément : longueur, module d'élasticité, moment quadratique.

Puis charges réparties ? (ici sur l'élément (4,5) on introduira par exemple −7N/mm).
[Ce problème sera repris par la suite.]

IV. TRAITEMENT

Si l'on procède comme ce qui a été vu antérieurement, on se retrouvera avec des inconnues dans 1 côté de l'équation matricielle et d'autres inconnues dans l'autre. Il faut donc classer les inconnues.

Prenons l'exemple suivant :
$1 \times x + 2 \times y + 3 \times z = 10$
$4 \times x + 5 \times y + 6 \times z = 11$
$7 \times x + 8 \times y + 9 \times z = 12$
ou

$$\begin{vmatrix} 10 \\ 11 \\ 12 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

Dans ce cas, il suffit de triangulariser la matrice et de ramener ce système à

$$\begin{vmatrix} A \\ B \\ C \end{vmatrix} = \begin{vmatrix} \alpha & B & \gamma \\ 0 & S & M \\ 0 & 0 & E \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

et résoudre pr z, y et enfin x.
Or, nous sommes dans le cas ou nous aurons :

$$\begin{vmatrix} 10 \\ y \\ 12 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \begin{vmatrix} x \\ 0 \\ z \end{vmatrix}$$

Il faut donc transposer y de l'autre côté de l'égalité. Dans ce cas général, on aura la valeur 0 pour y correspondant ; Pour la liaison 2 on aura -ky comme effort pour un déplacement de y

$10 = x + 2 \times 0 + 3z$
$y = 4x + 5 \times 0 + 6z$
$12 = 7x + 8 \times 0 + 9z$

$10 - 2 \times 0 = x + 0 + 3z$
$- 5 \times 0 = 4x + y + 6z$
$12 - 8 \times 0 = 7x + 0 + 9z$

d'où

$$\begin{vmatrix} 10 \\ 0 \\ 12 \end{vmatrix} = \begin{vmatrix} 1 & 0 & 3 \\ 4 & 1 & 6 \\ 7 & 0 & 8 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

Il suffit de mettre zéro dans la colonne de même indice que la ligne modifiée puis de mettre 1 sur le terme diagonal.

Cas de l'appui élastique

$$\begin{vmatrix} 10 \\ -ky \\ 12 \end{vmatrix} \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \begin{vmatrix} x \\ y \\ z \end{vmatrix}$$

$10 = x + 2y + 3z$
$-ky = 4x + 5y + 6z$
$12 = 7x + 8y = 9z$

$10 = x + 2y + 3z$
$0 = 4x + (5 + k) + 6z$
$12 = 7x + 8y + 9z$

Il suffit d'ajouter simplement le cœfficient k sur la diagonale correspondante.

V. RESOLUTION : TRIANGULATION

Traitons l'exemple suivant :

$$\begin{vmatrix} 10 \\ 11 \\ 12 \end{vmatrix} = \begin{vmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{vmatrix} \begin{vmatrix} z \\ y \\ z \end{vmatrix}$$

Afin de traiter ce système ; je rentre les noms des inconnues dans une matrice colonne de 3 éléments. D$ (0) = « x » ; D$ (1) = « y » ; D$ (2) = « z » ; et le système dans une seule matrice B (3,2). ATTENTION, la numérotation commence à Ø.

$$B\ (3,2) = \begin{vmatrix} 1 & 2 & 3 & 10 \\ 4 & 5 & 6 & 11 \\ 7 & 8 & 9 & 12 \end{vmatrix}$$

RESOLUTION : je garde la première ligne. A la place du 4 de la deuxième ligne, il faut faire apparaître un Ø en soustrayant 4 fois le premier terme.

$$\begin{vmatrix} 1 & 2 & 3 & 10 \\ 0 & -3 & -6 & -29 \\ 7 & 8 & 9 & 12 \end{vmatrix} = B\ (i,j)$$

Ensuite, il faut faire apparaître un Ø à la place du 7 [si on avait eu déjà un Ø ; on passe à la deuxième colonne directement (où à la suite)].

On a donc besoin d'une boucle.
FOR J = Ø TO R-1 et d'une boucle
FOR I = J + 1 TOR
Le calcul se fait sur toute la ligne par une troisième boucle FOR K = J TO R+1
La matrice est alors diagonalisée. Il faut alors repartir depuis le dernier terme pour mettre les résultats dans la colonne [F].

Remarque : Comme on ne dispose pas de l'indice − 1, il faut traiter seul le premier terme B(Ø,F) = B(Ø,F) / B(Ø,Ø)

VI. PRESENTATION

• Chaque tronçon étant isolé ; pour obtenir l'action du tronçon sur le nœud (hors des appuis), il faut changer le signe trouvé (lorsque D \$ = "F" ou "Mt").

• Lorsque l'on a un appui élastique, on obtient, en plus de α et V, la valeur de l'effort sur l'appui (on met "V." dans D \$ puis ensuite "V").

B(R,V) — D\$(R)

	Ø,Ø … (Ø,R)	2ᵉ moment (2*Z)	Données (S)	(U)	(V)	D\$(R)
Encastrement		0 0		Inertie Long	E(young) Q chap.	F M
App. élastique		0 0	K	I L	E Q	STR\$ K α
App. simple		0 Mt	Mt	I L	E Q	F α
Libre		F Mt	F Mt	I(i) L(i)	E(i) Q(i)	V α
	Matrice carrée (R*R) (R,Ø)					

Les résultats s'inscrivent dans cette colonne. F

Les 2ᵉ colonnes de B(R,V) sont ensuite utilisées sous la forme

F	V
M	α
⋮	⋮

VII. T, Mf, flèche

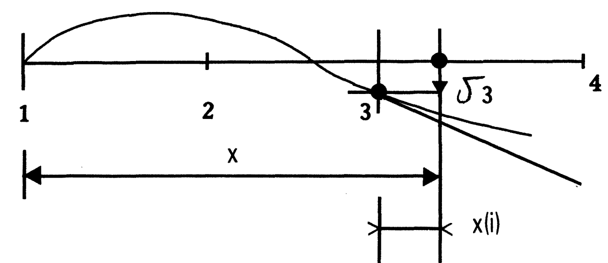
Ces premiers résultats étant donnés, on utilise les deux premières colonnes B(i,Ø) et B(i,1) pour mettre, en chaque nœud T,Mf,V,α. Ensuite on pourra connaître T, Mf et flèche en chaque point de la poutre. On se positionne à la cote X par rapport à l'origine de la poutre (nœud 1). le programme détermine dans quel tronçon se trouve le point demandé et on a :

$T = T(i) - x(i)\,Q(i)$

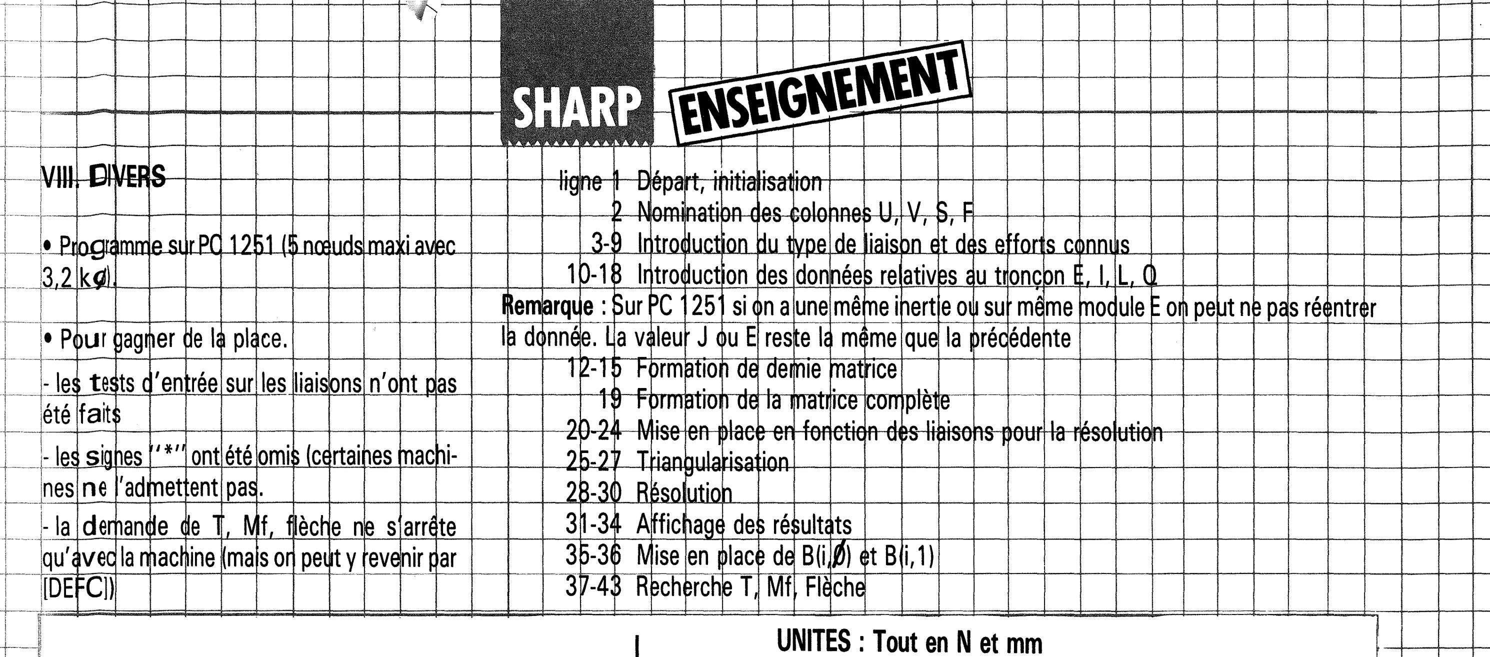
$Mf = Mf(i) - x(i)\,t(i) + Q(i)\,\frac{x(i)^2}{2}$

$\text{Flèche} = V(i) + \alpha(i)\,x(i) + T(i)\,\frac{x(i)^3}{6}$

$Mf(i)\,\frac{x(i)^2}{6} - \frac{Q(i)x(i)^4}{24}$



ici en 3, on a T(3) , Mf(3), V(3), α(3)



VIII. DIVERS

• Programme sur PC 1251 (5 nœuds maxi avec 3,2 kø).

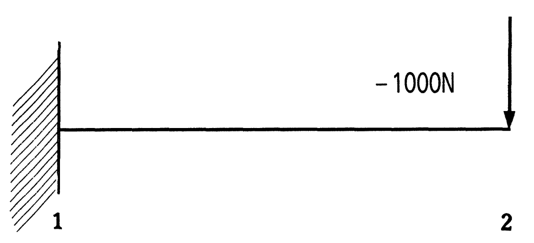
• Pour gagner de la place.

- les tests d'entrée sur les liaisons n'ont pas été faits
- les signes "*" ont été omis (certaines machines ne l'admettent pas.
- la demande de T, Mf, flèche ne s'arrête qu'avec la machine (mais on peut y revenir par [DEFC])

ligne 1 Départ, initialisation
2 Nomination des colonnes U, V, S, F
3-9 Introduction du type de liaison et des efforts connus
10-18 Introduction des données relatives au tronçon E, I, L, Q

Remarque : Sur PC 1251 si on a une même inertie ou sur même module E on peut ne pas réentrer la donnée. La valeur J ou E reste la même que la précédente

12-15 Formation de demie matrice
19 Formation de la matrice complète
20-24 Mise en place en fonction des liaisons pour la résolution
25-27 Triangularisation
28-30 Résolution
31-34 Affichage des résultats
35-36 Mise en place de B(i,Ø) et B(i,1)
37-43 Recherche T, Mf, Flèche



UNITES : Tout en N et mm

- Nombre de nœuds 2
- Nœud numéro 1 3
- Nœud numéro 2 0
- Tronçon 1
  - Force -1000
  - Moment 0
  - Longueur 400
  - Module E 2 E5
  - Moment quadratique $\pi \times (35 \wedge 4)$
  - Charges réparties N / 64

Résultats

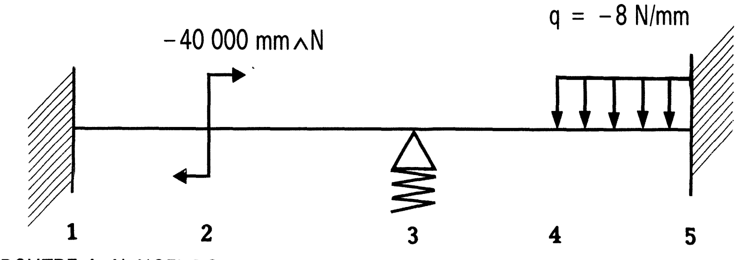
POUTRE A N NOEUDS
F1 = 1000.000001
M1 = 399999.9997
V2 = -1.4480603
@2 = -5.430226124E-03

Demande à x = 200

T, MF, FLECHE
T = 1000.000001
MF = 199999.9995
FL = -4.525188436E-01

Demande à x = 300

T = 1000.000001
MF = 9999.9994
FL = -0.916350658



Nombre de nœuds : 5

1 → 3
2 → 0 { Force 0 ; Moment -40 000 }
3 → 2 k = 2 000 N/mm
4 → 0 { Force = 0 ; Moment = 0 }
5 → 3

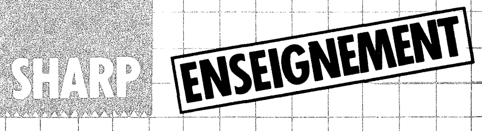
l(i) = 100 mm
Ei = 2 E 5 N/mm²
Ti = 100 000 mm⁴

POUTRE A N NOEUDS
F1 = -76.0080642
M1 = 12190.86022
V2 = -3.681115592E-03
@2 = -7.995631719E-05
F3 = 14.51612903
V3 = -7.258064515E-03
@3 = 2.083333324E-06
V4 = -4.41028226E-03
@4 = 4.974798386E-05
F5 = 861.4919356
M5 = -39690.86022

Demande à X = 250

T, MF, Flèche
T = -61.49193517
MF = -9532.930182
FL = -6.429981514E-03

```
1:"A" CLEAR : PRINT "-
  - POUTRE A  N NOEUDS
   --": INPUT "NOMBRE
  DE NOEUDS  ";Z:Y=Z-1
  :F=2Z:R=F-1
2:S=F+1:U=S+1:V=U+1:
  DIM B(R,V),D$(R)
3:FOR I=0 TO Y:M=2I:N=
  M+1: PAUSE "NOEUD NU
  MERO ("+ STR$ (I+1)+
  ")": INPUT "TYPE DE
  LIAISON  ";C
4:IF C=0 THEN INPUT "F
  ORCE  ";B(M,F),"MOME
  NT ";B(N,F):D$(M)="V
  ":D$(N)="@"
5:IF C=0 THEN LET B(M,
  S)=B(M,F):B(N,S)=B(N
  ,F)
6:IF C=1 THEN INPUT "M
  OMENT ";B(N,F):D$(M)
  ="F":D$(N)="@":B(N,S
  )=B(N,F)
7:IF C=2 THEN INPUT "C
  OEFF  K   ";T:D$(M)=
  STR$ T:D$(N)="@":B(M
  ,S)=T
8:IF C=3 THEN LET D$(M
  )="F":D$(N)="M"
9:NEXT I
10:FOR P=0 TO Y-1:
   PAUSE "TRONCON NUMER
   O("; STR$ (P+1);")":
   INPUT "LONGUEUR  ";L
   ,"MODULE ELAS. ";E
11:INPUT "MOMENT QUAD
   ";J
12:I=2P:K=2EJ/L:H=3K/L:
   G=2H/L:M=I+1:N=I+2:O
   =I+3
13:B(M,U)=L:B(I,U)=J:B(
   I,V)=E:B(R,U)=B(R,U)
   +L:B(I,I)=B(I,I)+G:B
   (I,M)=B(I,M)+H
14:B(I,N)=-G:B(I,O)=H:B
   (M,M)=B(M,M)+2K:B(M,
   N)=-H:B(M,O)=K:B(N,N
   )=B(N,N)+G
15:B(N,O)=B(N,O)-H:B(O,
   O)=B(O,O)+2K: INPUT
   "CHARGES REPARTIES ?
    ";A$: IF A$="N"
   THEN GOTO 18
16:INPUT "VALEUR  Q
   ";W:B(I,F)=B(I,F)+WL
   /2:B(M,F)=B(M,F)+WLL
   /12
17:B(N,F)=B(N,F)+WL/2:B
   (O,F)=B(O,F)-WLL/12:
   B(M,V)=-W
18:NEXT P
19:FOR I=0 TO 2Y: FOR J
   =I+1 TO R:B(J,I)=B(I
   ,J): NEXT J: NEXT I
20:FOR I=0 TO R: IF D$(
   I)="V" OR D$(I)="@"
   GOTO 24
21:IF D$(I)="F" OR D$(I
   )="M" GOTO 23
22:B(I,I)=B(I,I)+ VAL (
   D$(I)):D$(I)="V.":
   GOTO 24
23:FOR J=0 TO R:B(J,I)=
   0: NEXT J:B(I,I)=1
24:NEXT I
25:FOR J=0 TO R-1: FOR
   I=J+1 TO R: IF B(I,J
   )=0 GOTO 27
26:Q=B(I,J)/B(J,J): FOR
   K=J TO F:B(I,K)=B(I,
   K)-Q*B(J,K): NEXT K
27:NEXT I: NEXT J
28:FOR J=R TO 1 STEP -1
   : IF B(J,J)=0 BEEP 2
   : PRINT "EQUILIBRE I
   NDIFFERENT": END
29:B(J,F)=B(J,F)/B(J,J)
   : FOR I=J-1 TO 0
   STEP -1
30:B(I,J)=B(I,J)*B(J,F)
   :B(I,F)=B(I,F)-B(I,J
   ): NEXT I: NEXT J:B(
   0,F)=B(0,F)/B(0,0):
   BEEP 3
31:J=1.2: FOR I=0 TO R:
   IF D$(I)="F" OR D$(I
   )="M" LET B(I,F)=-B(
   I,F):B(I,0)=B(I,F):B
   (I,1)=0: GOTO 34
32:IF D$(I)="V." THEN
   LET B(I,S)=-B(I,S)*B
   (I,F):D$(I)="V":
   PRINT " F"+ STR$ INT
   J+" = ";B(I,S)
33:B(I,0)=B(I,S):B(I,1)
   =B(I,F)
34:PRINT " ";D$(I)+
   STR$ INT J;" = ";B(I
   ,F):J=J+.5: NEXT I
35:FOR I=2 TO R-2 STEP
   2:J=I-1:B(I,0)=B(I,0
   )+B(I-2,0)-B(J,V)*B(
   J,U)
36:B(I+1,0)=B(I+1,0)+B(
   J,0)+(B(J,V)*B(J,U)*
   B(J,U)/2)-(B(I-2,0)*
   B(J,U)): NEXT I
37:"C": PRINT " T , MF
   , FLECHE "
38:I=0: INPUT "CALCUL A
    X =  ";X: IF X>B(R,
   U) GOTO 38
39:M=2I:N=M+1: IF X<=B(
   N,U) GOTO 41
40:X=X-B(N,U):I=I+1:
   GOTO 39
41:G=B(M,0)-B(N,V)X:H=B
   (N,0)-B(M,0)X+B(N,V)
   XX/2:Q=B(M,U)*B(M,V)
42:K=B(M,1)+B(N,1)X-B(N
   ,0)XX/2Q+B(M,0)XXX/6
   Q-B(N,V)XXXX/24Q
43:PRINT " T  = ";G:
   PRINT " MF = ";H:
   PRINT " FL = ";K:
   GOTO 38
```



Fonction involute

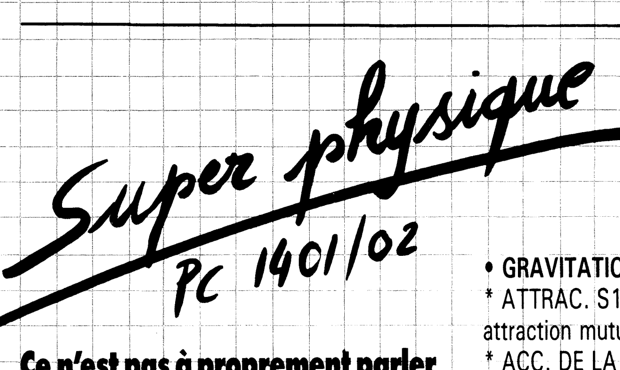
machine : SHARP PC 1261

La fonction involute Θest définie par :
inv Θ = tg Θ − Θ (avec Θ exprimé en radians)
Elle est utilisée couramment dans les calculs d'engrenages dont les dentures sont des arcs de développante de cercle.
Si trouver l'involute connaissant l'angle est très facile, le problème inverse ne l'est pas.
Il y a 10 ans, la méthode employée pour trouver Θ connaissant inv Θ consistait à se référer à des tables de valeurs, en effectuant éventuellement les interpolations nécessaires.
Le très court programme ci-joint remplace ces tables, avec une précision très supérieure (10^{-9}). Il utilise une méthode de calcul approché par itération, fondée sur le fait qu'entre 0° et 90° la fonction involute est toujours croissante — très lentement, bien qu'involute Θ tende vers l'infini si Θ tend vers $\pi/2$ radians. En effet, sa dérivée tg^2 Θ est toujours positive.
Les valeurs techniquement intéressantes de Θ pour le spécialiste de l'engrenage sont en fait comprises entre 10° et 50°. Il suffit d'entrer la valeur numérique de l'involute à la ligne 10 020 pour que le résultat s'affiche au bout d'un temps assez court à la ligne 10 090.

Daniel MAGNIN

Biographie : Traité théorique et pratique des engrenages par G. Henriot (Dunod éditeur).

```
10000:"I": REM Involute
10010:CLEAR : CURSOR 26:
      PRINT "Involute---
      >Theta": CLS
10020:CALL 39423: INPUT
      "INV Theta ? ";I:N
      =0:X=90: CLS :
      CURSOR 1
10030:T=(N+X)/2:J= TAN T
      -T*π/180
10040:IF ABS (J-I)<1E-9
      THEN 90+10000
10050:IF J>I THEN 70+100
      00
10060:N=T: GOTO 30+10000
10070:X=T: GOTO 30+10000
10090:BEEP 2: PAUSE DMS
      T: CURSOR 24:
      PRINT "Degres,mn,s
      "
10100:END
```



Ce n'est pas à proprement parler un seul et unique programme. Il s'agirait plutôt d'un recueil de 20 formules de physique.

UTILISATION

Une fois le pgm lancé par RUN, il vous propose un menu général :
• **Gravitation** équations quant à la gravitation (sur n'importe quelle planète).
• **Chute solide** équations relatives à la chute d'un solide
• **Lance Vertic** équations relatives au jet vertical d'un solide.
• **Loi de Stevin**
• **Loi d'Archimède**
• **Jet d'un solide** équations du jet d'un corps.
Pour choisir un titre affiché, pressez une touche alphanumérique quelconque puis ENTER, sinon pressez uniquement ENTER.
Chaque option renvoie à un sous-menu :

• **GRAVITATION ?**
* ATTRAC. S1/S2 ?
attraction mutuelle de 2 sphères.
* ACC. DE LA PES ?
calcul de l'accélération de la pesanteur.
* V. DE LIBER. ?
vitesse de libération
* SATELLITES
rayon de l'orbite géostastionnaire.
• **CHUTE SOLIDE ?**
* TMPS DE CHUTE
temps de chute.
* V. FINALE CHUTE
vitesse à l'arrivée au sol.
* ALTITUDE
calcul d'une altitude intermédiaire
* VITESSE
calcul d'une vitesse intermédiaire.
• **LANCE. VERTIC. ?**
* ALTI MAXI. ?
altitude maximale atteinte par le corps.
* TPS -) ALTI MAXI. ?
temps mis pour atteindre cette altitude.
* ALTI INTERNE. ?
calcul d'une altitude intermédiaire.
* VI. INTERNE. ?
calcul d'une vitesse intermédiaire.
• **JET D'UN SOLIDE ?**
* EQU. CARTES ?
calcul de l'ordonnée Y d'un point de la trajectoire par l'équation cartésienne.
* CALCUL DE Y
sans passer par l'équation cartésienne.
* ALTI. MAXI. ?
voir plus haut.
* TPS -) ALTI. MAXI ?
idem.
* TPS DE VOL
durée de vol de l'objet, du solide.
* PORTEE
distance parcourue par l'objet avant de retomber.
• **LOI DE STEVIN**
Calcul de la pression d'un liquide à une profondeur H.
• **LOI D'ARCHIMEDE**
Calcul de la poussée d'Archimède s'exerçant vers le haut (Euréka !)
Unités : Système MKSA
longueur = mètre ; temps = seconde.
masse = kilogramme ; pression = pascal (1 bar = 10^5 Pa)
densité = kilogramme/mètre ^ 3
pesanteur = mètre/seconde ^ 2
volume = mètre ^ 3
Bien entendu, vous pouvez répondre aux ques-

tions posées par le PC par des expressions qu'il calculera :
exemple : « M. PLANETE ? »
5.99 x 10 ^ 24 ENTER
Une fois le résultat affiché, le PC demande : « NOUVEAU CALCUL » répondez
• O pour recommencer la même routine.
• N pour revenir au menu. Voici la signification des variables :
G = pesanteur
M = poids
A = angle
R = distance
T = temps
V = vitesse (volume pour Archimède)
H = altitude (profondeur pour Stevin)
O = densité
PO = pression pour Stevin
Exemple de calcul :
Vous pouvez connaitre la pesanteur au niveau de la mer :
faire RUN
GRAVITATION ? tapez O ENTER
ACC. DE LA PES. ? tapez O ENTER
M. PLANETE tapez 5.99E24 (masse de la terre)
DISTANCE DU... CENTRE PLANETE ? tapez 6380E3 (rayon terre)
Affichage : 9.81988925 (m/S ^ 2)

A. OZANNE

```
1:"ALAIN OZANNE
2:"N. 41665
10:CLEAR
15:PRINT "    ";CHR$ 91;
   "PHYSIQUE";CHR$ 93
50:INPUT "GRAVITATION?"
   ;A$:GOTO 150
60:INPUT "CHUTE SOLIDE?
   ";A$:GOTO 200
70:INPUT "LANCE. VERTIC
   .?";A$:GOTO 240
80:INPUT "JET D UN SOLI
   DE";A$:GOTO 300
90:INPUT "LOI DE STEVIN
   ?";A$:GOTO 2400
100:INPUT "LOI D ARCHIME
   DE";A$:GOTO 2600
145:GOTO 50
150:INPUT "ATTRAC. S1/S2
   ?";A$:GOTO 500
160:INPUT "ACC.DE LA PES
   .?";A$:GOTO 600
170:INPUT "V. DE LIBER.?
   ";A$:GOTO 700
180:INPUT "SATELLITES ?"
   ;A$:GOTO 800
185:GOTO 150
200:INPUT "TMPS DE CHUTE
   ?";A$:GOTO 900
210:INPUT "V. FINALE CHU
   TE";A$:GOTO 1000
220:INPUT "ALTITUDE?";A$
   :GOTO 1100
230:INPUT "VITESSE?";O:
   GOTO 1200
235:GOTO 200
240:INPUT "ALTI MAX?";A$
   :GOTO 1300
250:INPUT "TPS->ALTI MAX
   ?";A$:GOTO 1400
260:INPUT "ALTI INTERME.
   ?";A$:GOTO 1500
270:INPUT "VI. INTERMED.
   ?";A$:GOTO 1600
275:GOTO 240
300:INPUT "EQU. CARTES.?
   ";A$:GOTO 1700
310:INPUT "CALCUL DE Y?"
   ;A$:GOTO 1800
320:INPUT "ALTI MAXI?";A
   $:GOTO 1900
330:INPUT "TPS->ALTI MAX
   I?";A$:GOTO 2000
340:INPUT "TPS DE VOL?";
   A$:GOTO 2100
350:INPUT "PORTEE?";A$:
   GOTO 2200
400:GOTO 300
490:"GRAVITATION
500:PRINT "ATTRACTION EN
   TRE":PRINT "2 SPHERE
   S S1&S2:"
510:INPUT "M DE S1?";M1
515:INPUT "M DE S2?";M2
520:INPUT "DIST. S1 A S2
   ?";R
525:F=6.673*10^-11*((M1*
   M2)/(R^2))
530:PRINT F
540:W=510:GOTO "Q"
600:PRINT "ACCELERATION
   DE":PRINT "LA PESANT
   EUR :"
610:INPUT "M. PLANETE?";
   M
615:PRINT "DISTANCE DU .
   ..":INPUT "CENTRE PL
   ANETE?";R
620:G=6.673*10^-11*(M/(R
   ^2))
630:PRINT G
640:W=610:GOTO "Q"
700:PRINT "VITESSE DE ..
   .":PRINT "...LIBERAT
   ION"
710:INPUT "M. PLANETE?";
   M
715:INPUT "RAYON PLANETE
   ?";R
720:VL=√(2*6.673*10^-11*
   (M/R))
730:PRINT VL
740:W=710:GOTO "Q"
800:PRINT "RAYON DE ..."
   :PRINT " L ORBITE ..
   .":PRINT "GEOSTATION
   NAIRE"
810:INPUT "M. PLANETE?";
   M
815:PRINT "PERIODE DE ..
   .":INPUT "ROTATION (
   TPS)?";T
820:R=(6.673*10^-11*(M*(
   T^2))/(4*(π^2)))^(1/
   3)
830:PRINT R
840:W=810:GOTO "Q"
890:"CHUTE /VIT. INIT. =
   0
900:PRINT "TEMPS DE CHUT
   E":PRINT "D UN SOLID
   E"
910:GOSUB "P"
915:GOSUB "H"
920:T=√((2*H)/G)
925:PRINT T
930:W=910:GOTO "Q"
1000:PRINT "VITESSE FIN
    ALE":PRINT "DE CHU
    TE D UN":PRINT "SO
    LIDE."
1010:GOSUB "P"
1015:GOSUB "H"
1020:V=√(2*G*H)
1030:PRINT V
1040:W=1010:GOTO "Q"
1100:PRINT "ALTITUDE EN
    COUR":PRINT "DE C
    HUTE."
1110:GOSUB "H"
1120:GOSUB "P"
1130:GOSUB "T"
1140:Y=H-((1/2)*G*(T^2)
    )
1150:PRINT Y
1160:W=1110:GOTO "Q"
1200:PRINT "VITESSE D U
    N":PRINT "SOLIDE E
    N COUR":PRINT "DE
    CHUTE."
1210:GOSUB "T"
1220:GOSUB "P"
1225:V=-G*T
1230:PRINT V
1240:W=1210:GOTO "Q"
1290:"LANCE VERTIC.
1300:PRINT "ALTITUDE MA
    XI":PRINT "ATTEINT
    E LORS":PRINT " D
    UN LANCEMENT":
    PRINT "VERTICAL."
1310:GOSUB "V"
1315:GOSUB "P"
1320:Y=(V^2)/(2*G)
1330:PRINT Y
```

```
1340:W=1310:GOTO "Q"
1400:PRINT "TPS MIS POU
     R":PRINT "ATTEINDR
     E ALTI.":PRINT " M
     AXI."
1410:GOSUB "V"
1415:GOSUB "P"
1420:T=V/G
1430:PRINT T
1440:W=1410:GOTO "Q"
1500:PRINT "ALTITUDE IN
     TER-":PRINT "MEDIA
     IRE."
1510:GOSUB "V"
1515:GOSUB "T"
1520:GOSUB "P"
1530:Y=(V*T)-(0.5*G*(T^
     2))
1540:PRINT Y
1550:W=1510:GOTO "Q"
1600:PRINT "VITESSE INT
     ER-":PRINT "-MEDIA
     IRE:"
1610:GOSUB "V"
1615:GOSUB "P"
1620:GOSUB "T"
1630:V1=V-(G*T)
1640:PRINT V1
1650:W=1610:GOTO "Q"
1690:"JET SOLIDE
1700:PRINT "EQUATION ..
     .":PRINT "CARTESIE
     NNE."
1710:GOSUB "A"
1720:GOSUB "P"
1730:GOSUB "V"
1740:INPUT "X  :?";X
1750:Y=(X*(TAN A))-((X^
     2)*(G/(2*(V^2)*((
     COS A)^2))))
1760:PRINT "Y:";Y
1770:W=1750:GOTO "Q"
1800:PRINT "CALCUL DE Y
     ":PRINT "Y=ORDONNE
     E"
1810:GOSUB "A"
1820:GOSUB "V"
1830:GOSUB "P"
1840:GOSUB "T"
1850:Y=(V*SIN A)*T-(0.5
     *G*(T^2))
1860:PRINT Y
1870:W=1810:GOTO "Q"
1900:PRINT "ALTITUDE MA
     XI."
1910:GOSUB "V"
1920:GOSUB "P"
1930:GOSUB "A"
1940:Y=(V^2)*((SIN A)^2
     )/(2*G)
1950:PRINT Y
1960:W=1910:GOTO "Q"
2000:PRINT "TPS POUR ..
     .":PRINT "ATTEINDR
     E ALTI.":PRINT "MA
     XI."
2010:GOSUB "V"
2020:GOSUB "A":GOSUB "P
     "
2030:T=(V*SIN A)/G
2040:PRINT T
2050:W=2010:GOTO "Q"
2100:PRINT "TEMPS DE VO
     L"
2110:GOSUB "V"
2115:GOSUB "P"
2120:GOSUB "A"
2130:T=2*(V*SIN A)/G
2140:PRINT T
2150:W=2110:GOTO "Q"
2200:PRINT "  PORTEE"
2210:GOSUB "V"
2220:GOSUB "A"
2230:GOSUB "P"
2240:X=((V^2)*(SIN (2*A
     )))/G
2250:PRINT X
2260:W=2210:GOTO "Q"
2390:"LOI STEVIN
2400:PRINT "LOI DE STEV
     IN"
2410:PRINT "PRESSION A
     LA...":INPUT "SURF
     ACE ?";P0
2420:INPUT "DENSITE DU
     LI.?";Q
2430:INPUT "PROFONDEUR"
     ;H
2440:INPUT "PESANTEUR?"
     ;G
2450:P=P0+(Q*G*H)
2460:PRINT P
2470:W=2410:GOTO "Q"
2590:"ARCHIMEDE
2600:PRINT "LOI D ARCHI
     MEDE":PRINT "DIRIG
     EE VERS LE":PRINT
     "HAUT."
2610:INPUT "DENSITE DU
     LI.?";Q
2620:INPUT "PESANTEUR?"
     ;G
2630:PRINT "VOLUME DU .
     ..":INPUT "CORPS I
     MMERGE?";V
2640:FA=Q*G*V
2650:PRINT FA
2660:W=2610:GOTO "Q"
65000:"V" INPUT "VIT. IN
      IT?";V
65005:RETURN
65010:"A" INPUT "ANGLE?"
      ;A
65015:RETURN
65020:"H" INPUT "ALTI?";
      H
65025:RETURN
65030:"T" INPUT "TEMPS(S
      EC)?";T
65035:RETURN
65040:"P" INPUT "PESANTE
      UR?";G
65045:RETURN
65100:"Q" INPUT "NOUVEAU
       CALCUL?";Z$
65110:IF LEFT$ (Z$,1)="O
      " GOTO W
65120:GOTO 50
```



SHARP ENSEIGNEMENT

PC Polynome

PC 1500

Après avoir entré le programme, il ne vous reste qu'à taper RUN.

Un menu s'affiche et vous choisissez la fonction désirée en tapant sur le chiffre correspondant. Le programme vous demandera les degrés des deux polynomes. Puis chacun des facteurs des puissances de X.

Cette opération terminée, le résultat s'affichera.

Les opérations disponibles sont :

Addition
Soustraction
Multiplication
Division (décroissante)
Division (croissante)

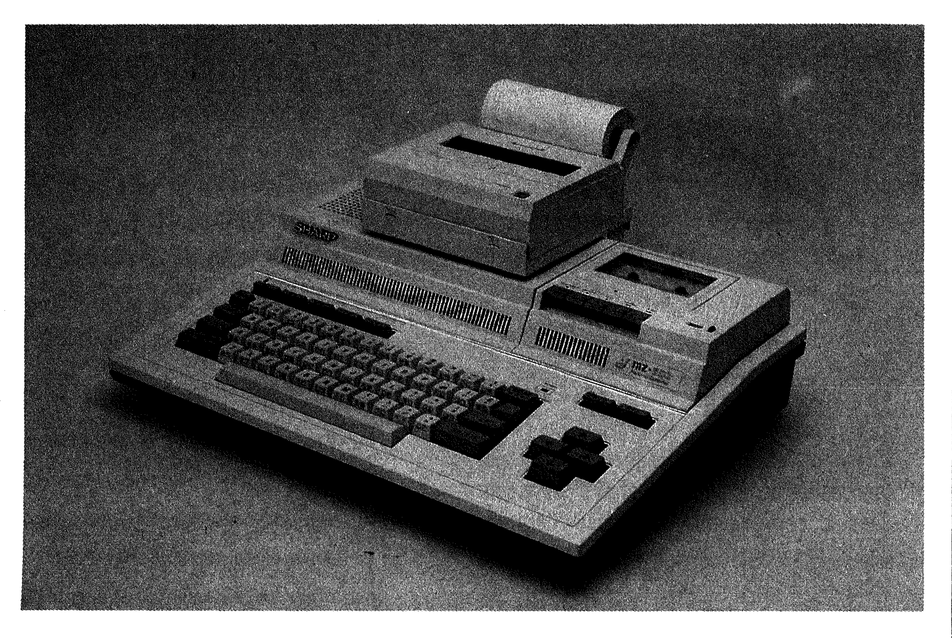
Ce programme peut aisément être adapté à tout autre PC ou MZ

```
10:"C"CLEAR :WAIT 0
20:PAUSE "******* POLYNOMES *******"
30:A$="Addition"
35:B$="Soustraction"
40:C$="Multiplication"
45:D$="Division (dec.)"
50:E$="Division (croi.)"
60:A=A+1-5*(A=5)
70:PRINT STR$ A;":";@$(A):BEEP 1,1,1
80:D=0:FOR I=0TO 15:F=VAL INKEY$ :IF F>0AND F<6LET D=1:I=20
90:NEXT I:IF D=0THEN 60
95:PAUSE @$(F)
100:INPUT "DEGRE DE A=";M
110:INPUT "DEGRE DE B=";N
120:GOSUB STR$ F:GOTO "C"
130:"E"V=-1:FOR I=0TO M:CLS :PRINT "a(";STR$ I;")="
140:CURSOR 6:INPUT A(I):IF A(I)<>0AND V=-1LET V=I
150:NEXT I
160:W=-1:FOR I=0TO N:CLS :PRINT "b(";STR$ I;")="
170:CURSOR 6:INPUT B(I):IF B(I)<>0AND W=-1LET W=I
180:NEXT I:RETURN
190:"A2"CLS :WAIT :PAUSE "RESULTAT":BEEP 1:CLS
195:FOR I=XTO Y:PRINT "c(";STR$ I;")=";C(I):NEXT I:RETURN
200:"1"T=N+(M-N)*(M>N)
210:DIM A(T),B(T),C(T):GOSUB "E"
220:FOR I=0TO T:C(I)=A(I)+B(I):NEXT I
230:Y=T:X=V-(W-V)*(W<V):GOTO "A2"
240:"2"T=N+(M-N)*(M>N)
250:DIM A(T),B(T),C(T):GOSUB "E"
260:FOR I=0TO T:C(I)=A(I)-B(I):NEXT I
270:Y=T:X=V-(W-V)*(W<V):GOTO "A2"
280:"3"DIM A(M),B(N),C(M+N):GOSUB "E"
290:FOR I=0TO M:FOR J=0TO N
300:C(I+J)=C(I+J)+A(I)*B(J):NEXT J:NEXT I
310:Y=M+N:X=V+W:GOTO "A2"
320:"4"IF N>MWAIT :PRINT "Deg(B)>Deg(A):IMPOSSIBLE":RETURN
330:DIM A(M),B(N),C(M-N)
340:GOSUB "E"
350:FOR I=M-NTO 0STEP -1:C(I)=A(I+N)/B(N)
360:FOR J=NTO 0STEP -1:A(I+J)=A(I+J)-C(I)*B(J)
370:NEXT J:NEXT I
380:Y=M-N:Z=M-N:GOTO "A1"
390:"5"INPUT "ORDRE=";S
400:T=M:IF T<N+SLET T=N+S
410:DIM A(T),B(N),C(S)
420:GOSUB "E":IF V<=WGOTO 430
425:WAIT :CLS :PRINT "Val(A)<Val(B):IMPOSSIBLE":RETURN
430:FOR I=0TO S:C(I)=A(I)/B(0)
440:FOR J=ITO I+N:A(J)=A(J)-C(I)*B(J-I)
450:NEXT J:NEXT I
460:Y=S:Z=M:IF Z<S+NLET Z=S+N
470:T=V+S:GOTO "A1"
500:"A1"CLS :WAIT :PAUSE "QUOTIENT":BEEP 1
510:FOR I=0TO Y:PRINT "q(";STR$ I;")=";C(I):NEXT I
520:PAUSE "RESTE":BEEP 1
530:FOR I=TTO Z:PRINT "r(";STR$ I;")=";A(I):NEXT I
540:RETURN
```

Z = N-1


```
10 input F   B=1  C=1  D=Ø   G=F   E=INT(F)   A=E
20 IF ABS( F- A/B) < 10E-6  PRINT A;"/";B : END
30 G=1/(G-E) : E=INT(G)  H= A*E+C  C=A  A=H
   H= B*E+D  D=B  B=H  GOTO 20
```

MZ-800 LE PLUS PROFESSIONNEL DES INDIVIDUELS



La machine a été conçue pour être la plus évolutive du marché. De 2500 F. pour la version de base à moins de 10 000 F. pour la configuration Unité Centrale + Clavier + Moniteur + Contrôleur + Unité de disquettes + Système d'exploitation CP/M.

Compatible avec les logiciels de son prédécesseur, le MZ 700 dont il a un mode d'émulation, il possède, en plus des logiciels éducatifs, des outils de développement et des langages en grand nombre.

Le MZ 800 est, sous CP/M une machine aux performances étonnantes. Les Best-sellers du marche CP/M sont implantés et disponibles chez nos distributeurs : **MULTIPLAN, WORDSTAR, MAILMERGE, CUPER CALC 2, DBASE 2,** ainsi que de nombreux logiciels « verticaux » : **Paie, compta, facturation,** etc. N'ont pas été oubliés les logiciels de communications : **BSTAM** et **BSTMS**, ni les langages : **BASIC MICROSOFT, BASCOM, FORTRAN, PASCAL, Langage C, FORTH, MICRO LISP,** etc.

Le système d'exploitation lui-même, appelé **Personal CP/M** a été complètement repensé par SHARP. Il est articulé autour du programme superviseur appelé **VCCP** qui présente toutes les commandes résidentes du système sous forme de **menu.** Il suffit alors de se déplacer avec le curseur pour appeler la commande désirée. Une fonction **« HELP »** est disponible à tout moment, elle permet de renseigner l'opérateur sur toutes les possibilités et les syntaxes spécifiques de la commande choisie. Les utilitaires système ont été parfaitement adaptés aux spécificités du MZ 800 et permettent, entre autres :

- la définition de **32 touches de fonctions.**
- Le paramétrage de **2 interfaces RS 232 C**
- Le choix des couleurs de fonds et d'encre.

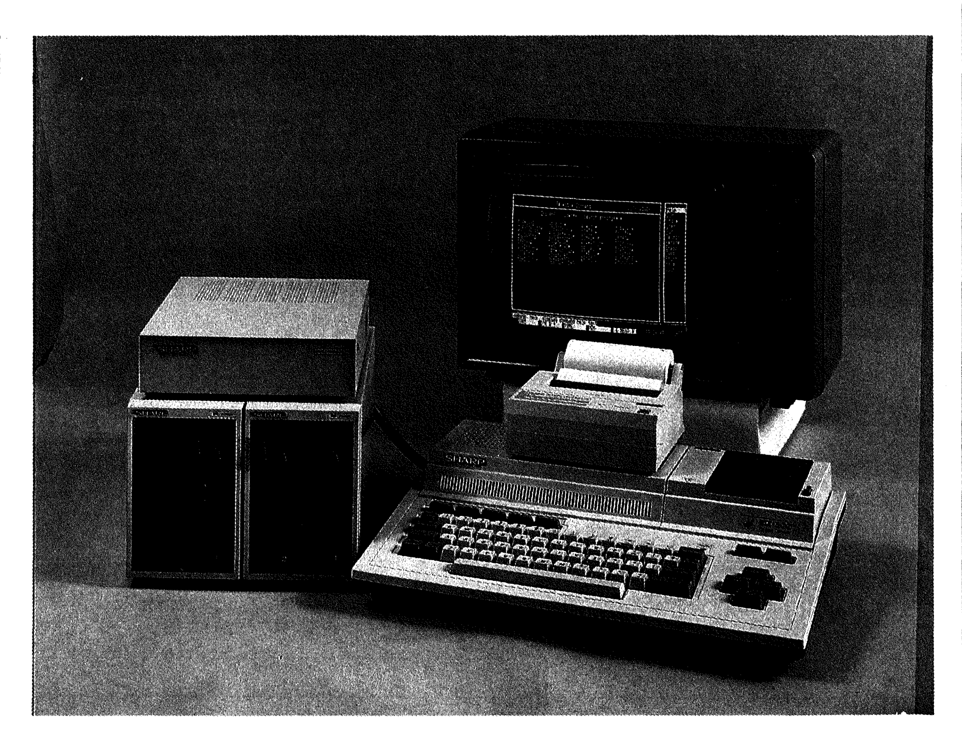
etc.

CARACTERISTIQUES TECHNIQUES

PRESENTATION COMPACTE

Dans sa version de base avec lecteur de cassettes incorporé, intégrant :

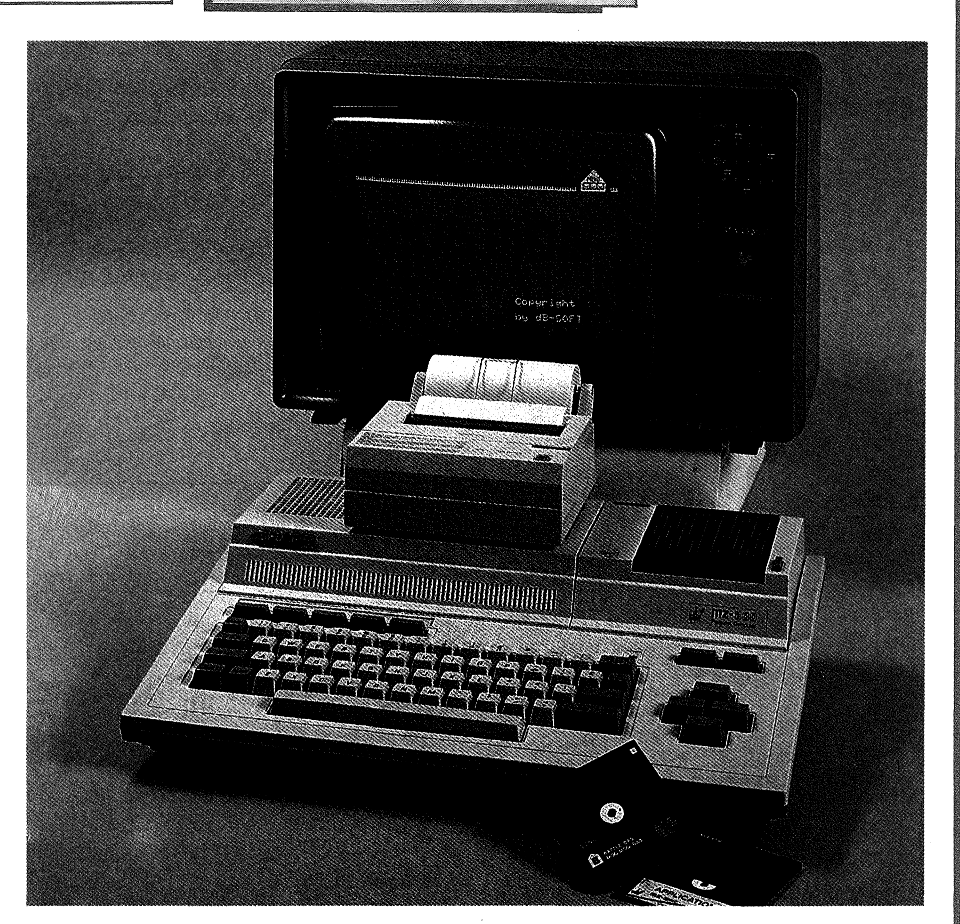
- Microprocesseur Z80A à 3,5 MHz.
- **16 Ko.** de ROM
- **64 Ko.** de RAM extensibles à 128 Ko.
- **16 Ko.** de RAM graphique INDEPENDANTE extensibles à 32 Ko.
- Clavier 69 touches
- Panier d'extensions
- Sortie parallèle **CENTRONICS**
- Prise **moniteur** R.G.B.
- Prise **PERITEL**
- Connecteurs pour 2 **JOYSTICKS compatibles ATARI**
- Générateur sonore programmable **3 voies / 8 octaves.**



ET EN PLUS

Les nombreuses options permettent de personnaliser le MZ 800 au gré des utilisations spécifiques de chacun :

- Ecran couleurs
- **Extension de mémoire graphique** autorisant 4 couleurs simultanées sur l'écran en résolution 640 x 200 points (2 en version de base) et 16 couleurs simultanées en 320 x 200 (4 de base).
- **Extension RAM 64 Ko.** permettant d'émuler une disquette virtuelle et de créer une mémoire tampon pour l'imprimante
- **QUICK DISC** pour l'utilisation des disquettes 2,8 pouces d'une capacité de 128 Ko.
- Unités de **disquettes 5 pouces 1/4** d'une capacité de 320 Ko. formatée chacune. Le MZ 800 peut gérer simultanément jusqu'à 4 de ces unités
- Carte de communication RS 232 C à 2 ports : 1 port synchrone/asynchrone + port asynchrone
- Imprimante table traçante 4 couleurs au format **11 cm**
- Imprimante table traçante 4 couleurs au format **21X27 cm**
- Langages sur cassette : **LOGO** (Digital Research), **MACRO-ASSEMBLEUR, PASCAL.**



CP/M

Revenons à CP/M. Les logiciels standard tournant sous ce système d'exploitation étant maintenant très connus nous étudierons plutôt le système lui-même. Conçu par DIGITAL RESEARCH et adapté spécialement sur MZ 800, PERSONAL CP/M est une réponse aux différentes critiques émises quant à l'utilisation du système précédent : CP/M 2.2. D'un maniement très simplifié, il est accessible même aux plus néophites. La différence est flagrante dès le démarrage du système. L'ancienne version, après une ligne de présentation ésotérique affichait un « A) » laissant l'utilisateur dubitatif quant aux commandes qu'il pouvait ou devait utiliser. Au contraire, PCP/M présente, dès la mise en route toutes les fonctions et programmes disponibles dans 2 fenêtres. Dans celle de droite : les commandes résidentes intégrées au système, dans celle de gauche : les commandes non résidentes et les programmes utilisateurs. La sélection de l'un d'entre eux s'effectue uniquement grâce aux quatre touches curseur. Signalons, pour conclure, que PCP/M est compatible à 100% avec la version 2.2 et que tout utilisateur pourra trouver un logiciel répondant à ses besoins dans l'immense bibliothèque existant sous ce système.

S.CALGO

Logiciel sur QUICK DISC ne signifie pas pour autant « logiciel pour amateur ». Rapide, performant et coloré, S.CALGO permet d'établir rapidement une feuille de calculs et de sauvegarder son travail soit sur cassette, soit sur micro-disquette. Ce programme séduit immédiatement par sa simplicité d'utilisation. Les commandes sont entrées essentiellement par les touches de fonctions et le curseur. S'il n'a pas tout à fait l'ergonomie de MULTIPLAN, il en a toutes les potentialités. Sa facilité d'utilisation, la rapidité d'éxécution de ses calculs et la qualité des résultats obtenus la placent à niveau égal avec toutes les feuilles de calculs electroniques de haut niveau. Son dernier atout, et non des moindres, reste son prix sans aucune commune mesure avec ceux couramment pratiqués pour des programmes équivalents.

nom du progiciel :
MULTICOMPTA V.M.
domaine d'application : COMPTABILITE
description :
PLAN COMPTABLE 82
• Création modification
• Visualisation édition
SAISIE DES ECRITURES par journaux avec
• ventilation
• libellés libres ou automatiques
MISE A JOUR ECRITURES
EDITION :
• Balances • Grand livre • Journaux
• Comptes de résultats • Echéancier
• Relances
UTILITAIRES :
• Société • Changement période et exercice
• Table des comptes • Fichier de saisie
matériel : MZ 800 + disquette et imprimante
système : CP/M
Langage : Basic Compile

---

nom du logiciel :
MULTI PAIE BATIMENT V.M.
domaine d'application : PAIE
description :
• Paramétrage des rubriques calculées avant/après brut
• Tenue des fiches salariés et des cumuls
• Tenue des données entreprise
• Saisie contrôlée et édition des bulletins de paie
• Annulation de paie
• Edition des états cumuls de paie
• Edition de la liste des virements
• Edition du D.A.S. individuel
• Edition du D.A.S. fiscal
• Edition attestation 1 200 h
• Edition des bordereaux individuels congés payés
• Edition états récapitulatifs congés payés
• Edition état récapitulatif C.N.R.O.
matériel : MZ 800 + disquette + imprimante
système : CP/M
langage : Basic compile

---

nom du logiciel : **FACTURATION / STOC V.M.**
domaine d'application : GESTION
description :
GESTION DE FICHIER CLIENTS :
• Création • Modifications • Annulation
• Liste • Surveillance découvert
GESTION DU FICHIER ARTICLES :
• Création • Modifications
• Annulation • Liste • 3 tarifs / article
CONTROLE DE L'INVENTAIRE :
• Liste des articles en rupture de stock
• Valorisaiton de l'inventaire
FACTURATION :
• En liaison avec les fichiers clients articles
• Adresse facturation et livraison
• Remise à la ligne escompte
• Port numéro de B.L. numéro de commande
• Mode expédition/mode règlement
• Validation
JOURNAL DES VENTES
GESTION DES VENTES
• Etats journalier, mensuel, périodique
OPTION : gestion des commandes en cours
matériel : MZ 800 + disquette + imprimante
système : CP/M
langage : Basic compile

---

nom du logiciel :
GESTION PARC ROULANT V.M.
domaine d'application : TRANSPORTEURS
description :
FACTURATION ET COMPTABILITE CLASSIQUE
GESTION DU PARC ROULANT
• Détermination du prix du km roulant et réel et du prix du km roulant commercial
• Modification de la tarification client
GESTION DES BORDEREAUX DE TOURNEE
matériel : MZ 800 + disquette + imprimante
système : CP/M
langage : Basic compile

---

nom du logiciel :
FACTURIERE TIERS PAYANT
domaine d'application : PHARMACIE
description :
GESTION FICHIERS
• Médecins nom numéro SS
• Mutuelles nom taux bleu et blanc
Saisie des coordonnées des malades
Saisie de la facturation
Contrôle
Edition de la facture subrogatoire
Edition du récapitulatif par mutuelle
volumes : 150 mutuelles 200 médecins
matériel : MZ 800 ou 700 cassette + imprimante feuille
langage : Basic compile
remarque : possibilité d'évolution vers la disquette pour le fichier client sur MZ 800

---

nom du progiciel :
VIDEO CLUB (I.M.F.).
domaine d'application : COMMERCE
description :
QUALITES DU SYSTEME VIDEO I.M.F.
SIMPLE :
N'utilise que 7 touches de fonctions
Permet de visualiser toutes les opérations
EFFICACE :
Emmagasine sur disquette toutes les informations
Plus de 100 listing à éditer au choix.
Entièrement paramétrable.
IL GERE :
Les locations, les restitutions, les adhérents
Les cautions, les réservations, le stock
La caisse, les fournisseurs, l'historique client
IL ANALYSE :
La rentabilité des films, la fréquence des adhérents, la rotation des films, l'usure des cassettes, les statistiques générales, les meilleures locations du mois ou de l'année.
IL CONTROLE :
Le chiffre d'affaires par film et par client, les sorties de films, les échéances d'adhésion.
IL EDITE :
Les bons de locations, les contrats d'adhésion, les catalogues, les étiquettes pour le courrier, les listes de films des clients, les factures, ect.
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile

---

nom du logiciel : **PHARMA (I.M.F.)**
domaine d'application : SANTE
description :
ORDONNANCE :
Sans délégation de paiement :
• Saisie des produits, prescripteurs, malade
• calcul automatique et édition de la feuille de Sécurité Sociale
Avec délégation de paiement :
• Saisie client, caisse, mutuelle, prescripteur
• Calcul automatique montant par caisse, mutuelle assuré, réductible, pansement
• Edition facture subrogatoire, feuille S.S.
• Relevé organisme tiers payant
Suivi remboursement :
• Liste du dossier en attente par client, par organisme, etc.
GESTION DES ACHATS : disque dur obligatoire
• Gestion des lignes lourdes répartiteurs et des spécialités aux tableaux, stock d'alerte, opportunités de commande, calcul des remises par répartiteur
COMPTABILITE DES VENTES : disque dur obligatoire
• Chaque soir édition d'un récapitulatif des ventes avec statistiques
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile

nom du progiciel :
CABINET DENTAIRE (I.M.F.)
domaine d'application : SANTE
description :
TRAITEMENT DES INFORMATIONS MEDICALES
Permet de gérer toutes les informations dont le praticien a besoin
FICHE SIGNALETIQUE DU PATIENT
Etat civil du patient, antécédents, mâchoire
L'historique des interventions apparaît à l'écran et peut être actualisé à tout moment.
FICHE MEDICALE DU PATIENT
le praticien dispose de tous les soins effectués pour chaque client. Le dessin de la mâchoire se modifie en fonction des soins appliqués.
GESTION DU CARNET DE RENDEZ—VOUS
Gestion indépendante qui permet au praticien de tenir le carnet de rendez-vous à jour.
ETATS :
Plus de 30 états à disposition
FICHIERS :
Dossiers - Actes médicaux
Clients - Contre indications
Règlements
LIAISON AVEC LE LOGICIEL COMPTABILITE
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile

---

nom du progiciel : **MAILING (I.M.F.)**
domaine d'application : COMMERCE TEXTES
description :
GESTION DE FICHIER CLIENT
• Création
• Modification
• Recherche
• Listes
La recherche multi-critères se fait sur toutes les informations du fichier.
Le module d'extraction d'adresses multi-critères permet de générer un fichier adresse et un fichier textes compatible WORDSTAR et MAILMERGE.
LIAISON A LA COMPTABILITE FACTURATION...
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile

---

nom du progiciel :
CABINET MEDICAL (I.M.F.)
domaine d'application : SANTE
description :
FICHIERS :
DOSSIERS PATIENTS RENDEZ-VOUS MEDECINS MEDICAMENTS DICTIONNAIRE
Il permet :
• Gestion des rendez-vous 1 ou plusieurs médecins
• Edition des rendez-vous de la journée avec extrait du dossier patient
• Enregistrement des observations dans le dossier parient (caractère médical ou non).
• Création des liens entre les observations (ex. cancer du poumon et lieu de travail enfumé)
• Création de l'ordonnance
• Edition de l'historique des patients
• Edition de formulaires de S.S.
• Gestion financière du cabinet
LIAISON AVEC LE LOGICIEL COMPTABILITE
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile

---

nom du progiciel :
GESTION METHODES DE VENTE (I.M.F.)
domaine d'application : COMMERCE
description :
Le logiciel MULTIVENTE permet de gérer plusieurs méthodes de vente
• Couponning (presse ou distribution)
• Mailing
• Représentants
Les coupons réponses ou coupons commandes sont traités dans deux modules différents
COUPONS REPONSES
• Affectation à un représentant
• Suivi du retour coupon
• Contrôle représentant
• Statistiques coupon
• Statistique représentants
COUPONS COMMANDES
• Confirmation commande
• Bons de livraison
• Facture
• Gestion du mode de paiement (crédit, comptant, etc.)
• Gestion des stocks
• Statistiques
• Editions automatiques SERNAM, PTT...
LIAISON A LA COMPTABILITE
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile

---

nom du progiciel :
BAR RESTAURANT STATISTIQUES (I.M.F.)
domaine d'application : TOURISME
description :
GESTION DU BAR
• Gérer les stocks en quantité et en valeur
• Contrôler les doses vendues
• Gérer les ventes
• Emission tickets de caisse, factures, relevés hebdomadaire ou mensuels, enregistrement des règlements
GESTION DU RESTAURANT
• Gestion des stocks de marchandises
• Enregistrement des fiches culinaires, ingrédients, temps préparation, temps cuisson..
• Gestion de la clientèle, réservations, mailings
• Emission des tickets de caisse ou des factures
• Gestion de l'équipe de restauration
STATISTIQUES
• Tableau de bord synthétiques des activités bar et restaurant
• Stock clientèle, repas
• Etude des prix de revient et des marges
LIAISON COMPTABILITE FACTURATION
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile

---

nom du progiciel :
GESTION AFFAIRES (I.M.F.)
domaine d'application : COMMERCE
description :
GESTION DE FICHIERS :
• Fichier des dossiers
Plusieurs dossiers sur une même affaire
• Fichier des prévisions
Enregistrement sur un dossier des tâches à effectuer et de leur coût prévu
• Fichier des réalisations
Enregistrement sur un dossier de la réalisation des tâches et de leur coût
• Fichier des tâches
Dictionnaire des tâches
Ce logiciel permet de suivre l'évolution d'une affaire et de préparer la facturation
LIAISON AGENDA et FACTURATION
matériel : MZ 800 + disquette * 2
système : CP/M
langage : DBASE II Version 2,4 Compile





Le PC 1211 est encore à l'honneur. Cette machine, en tant que doyenne de la longue lignée des PC, est évidemment très vénérée des Sharpentiers. Bien sûr, les mauvaises langues disent qu'elle est trop lente, et ils n'ont pas tout à fait tort...

Alors les Sharpentiers relèvent le gant : voici comment donner des ailes à votre PC 1211.

Il s'avère qu'en court-circuitant le quartz du PC, on force l'horloge à osciller plus vite qu'elle ne le fait sous le contrôle de ce quartz de 256 KHz.

Ainsi dopée, la vitesse d'exécution de la machine s'accroît d'environ 1,75 (n'ayez aucune crainte, ce n'est pas le maximum !). Cela présente un avantage indéniable pour les calculs à itérations.

Afin de pouvoir relire vos cassettes préférées, il est conseillé de pouvoir commuter de la vitesse lente à la vitesse rapide à tout moment. On confiera ce rôle à une ampoule I.L.S. Sous l'action d'un aimant, votre machine s'exécute à vitesse normale ou accélérée suivant que le contact de l'I.L.S. est fermé ou ouvert.

Les schémas vous montrent clairement l'installation de cette ampoule dans votre machine. Seul le démontage du couvercle inférieur du PC est nécessaire. Les bornes du quartz sont alors bien apparentes.

1500-istes et 1251-istes : préparez vos fers à souder, nous installerons ce turbo dans vos machines dans le prochain numéro.

C. POULAIN

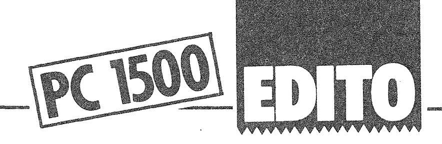
SC 43178

bornes du quartz

ampoules I.L.S.

interrupteur à lame souple

logement conseillé pour l'I.L.S.



Vos PC ont à peine digéré les programmes de notre précédent bulletin qu'en voici déjà de nouveaux, plein d'astuces et de découvertes. En ce début d'année scolaire, nous vous indiquons les textes, actuellement en vigueur, réglementant l'emploi des « calculatrices » aux examens :

2 octobre 1979 — B.O. N° 79318
28 octobre 1979 — B.O. N° 79409
10 novembre 1983 — F.P. N° 1535

Ces différents textes sont disponibles au Service de Documentation de l'Education Nationale, 13 rue Dufour 75006 Paris.

Vous trouverez, dans ce bulletin, de nouveaux produits, donc de nouvelles colonnes en perspective ; beaucoup de Basic, un peu de L.M. et des applications sur tableur. En attendant de nouvelles évolutions, voici notre orientation actuelle.

Luc BURELLER

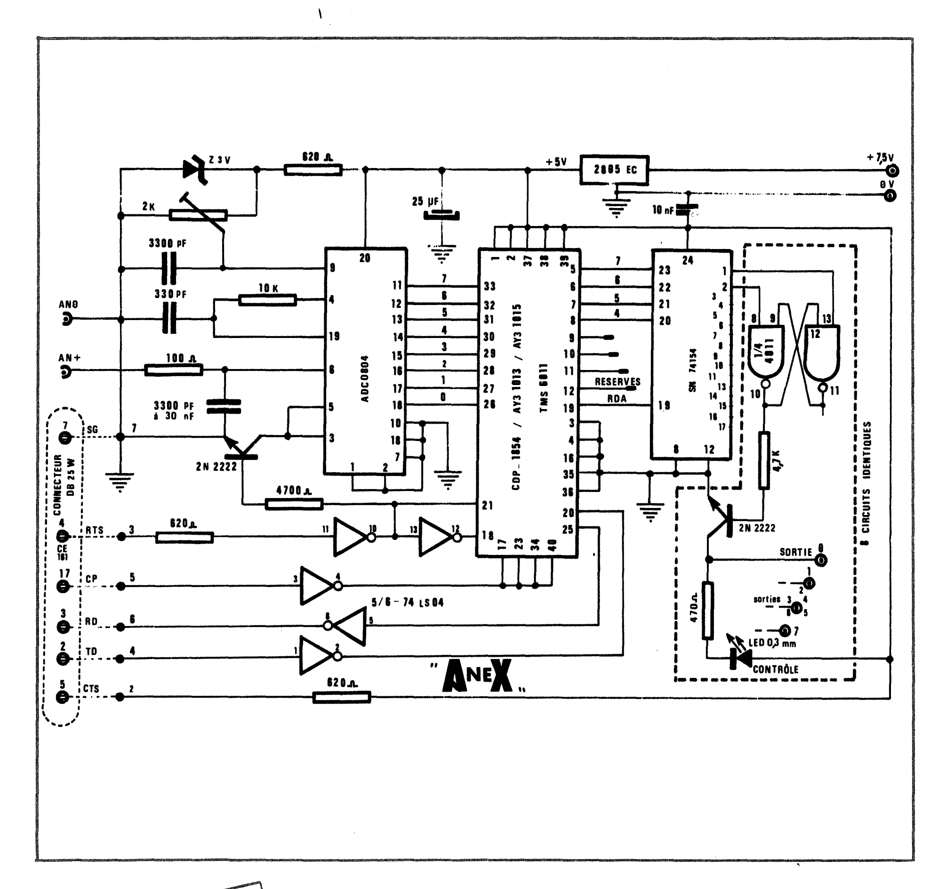


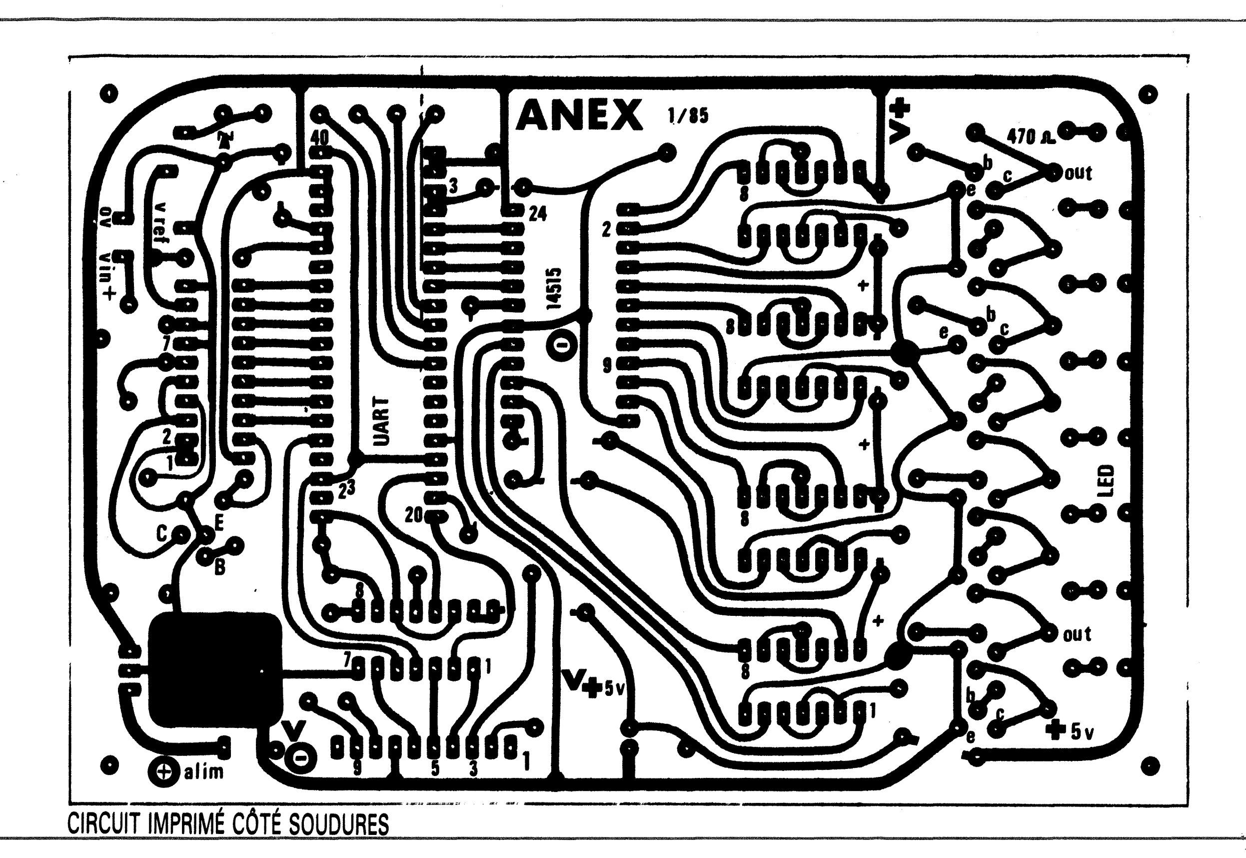
ANEX

ANEX POUR PC 1500

Hormis les appareils professionnels d'un coût prohibitif, rares sont les multimètres interfacés DCE V-24, pourtant ce ne sont pas les modèles de contrôleurs qui manquent ! Comme pour les micros on peut en compter trois nouvelles variantes chaque mois. Mais vraiment rien en RS 232C, il faudra donc encore attendre pour trouver les nouveaux musts sur les étals des silicon stores de faubourg.

Anex, acronyme pour analogique extension, n'est pas à proprement parler un multimètre, mais plutôt une interface de mesure doublée d'un port de sortie, compatible V-24. Coût de l'opération, moins de 300 F, plus éventuellement la perte de garantie sur la CE 158, qu'il faudra démonter pour y effectuer... deux soudures ! A la mise au point un multimètre disons, courant, conviendra pour peaufiner l'unique réglage.





CIRCUIT IMPRIMÉ CÔTÉ SOUDURES

```
 8 "AneX"REM      *******************************
10 REM            *  ROUTINE DE DEMONSTRATION  *
12 REM  OBJET     * SET SUR LES PORTS DE 0 à 7 *
14 REM            *     RESET ET BOUCLAGE      *
16 REM            *******************************
18 REM
20 SETCOM 2400,8,E,1:OUTSTAT 0:SETDEV PO
22 "ANX"LPRINT CHR$ 1;:REM  ACTIVATION VOIE 0
24 LPRINT CHR$ 65;:REM  ACTIVATION VOIE 1
26 LPRINT CHR$ 33;:REM  ACTIVATION VOIE 2
28 LPRINT CHR$ 97;:REM  ACTIVATION VOIE 3
30 LPRINT CHR$ 17;:REM  ACTIVATION VOIE 4
32 LPRINT CHR$ 81;:REM  ACTIVATION VOIE 5
34 LPRINT CHR$ 48;:REM  ACTIVATION VOIE 6
36 LPRINT CHR$ 113;:REM ACTIVATION VOIE 7
38 REM        * LES 8 SORTIES SONT ACTIVEES *
40 LPRINT CHR$ 129;:REM          RAZ VOIE 0
42 LPRINT CHR$ 193;:REM          RAZ VOIE 1
44 LPRINT CHR$ 161;:REM          RAZ VOIE 2
46 LPRINT CHR$ 225;:REM          RAZ VOIE 3
48 LPRINT CHR$ 145;:REM          RAZ VOIE 4
50 LPRINT CHR$ 209;:REM          RAZ VOIE 5
52 LPRINT CHR$ 177;:REM          RAZ VOIE 6
54 LPRINT CHR$ 241;:REM          RAZ VOIE 7
56 REM                 * RAZ GENERAL *
58 REM            SUIVANT CET EXAMPLE, D/AUTRES
60 REM            CODES ASCII SONT UTILISABLES
62 REM            ET PEUVENT ACTIONNER PLUSIEURS
64 REM            SORTIES A LA FOIS.
66 GOTO "ANX"
68 REM
70 "TELEM"REM     ******************************
72 REM            *     DEMO SAISIE MESURE     *
74 REM            ******************************
76 REM
78 SETCOM 2400,8,E,1:SETDEV KI:WAIT 0
80 GOSUB "RESET"
82 "TEL":A$=RINKEY$ :A=ASC A$:PRINT A;
84 IF INKEY$ GOSUB "RESET"
86 IF INSTAT =4GOTO "ALIMOFF"
88 GOTO "TEL"
90 "RESET"BEEP 1:OUTSTAT 2:OUTSTAT 0:RETURN
94 "ALIMOFF"BEEP 1:PRINT " ALIM/OFF"
96 GOTO "TELEM"
98 REM
99 REM            ******************************
```

UN UART EN CACHE TOUJOURS UN AUTRE !

Une fois encore nous retrouvons un ci à succès, l'Uart de Général Instrument, au cœur de notre carte de télémesure, il sérialise les données issues d'un convertisseur an, et inversement, restitue en mode parallèle les caractères reçus série, Le can (convertisseur analogique/numérique), est un 8 bits, ayant 256 états possibles, et ce, dans une plage de 0 à 2,56 V.

Le can compare par approximations successives, la tension à mesurer (broche 6) a une tension de référence (broche 9), pour ensuite, en verrouiller le résultat sur son bus (broches 11 à 18). Prévoir un pont diviseur pour les mesures supérieures à 5 volts DC. Dans la version présente les capteurs résistifs s'adaptent directement. Thermistances, photo résistances shunts pour mesure de courant et traçage de courbes consommation, tachymètre, manche à balai, voir bio feed back. C'est évidemment dans la mémorisation et le traitement des mesures que réside tout l'intérêt du système, par exemple, détection des mini/max, et action sur le port de sortie.

Naturellement les fonctions Telecom de la CE-158 permettent le dialogue avec Anex. Rinkey en setdev KI pour les entrées, et LPRINT sur setdev PO en sortie. Notez que 2400 bauds étant le maximum, PEEK et POKE n'amèneront pas de gain de temps spectaculaire.

COINCES PAR LEURS PROBLEMES DE COMMUNICATION

L'état des sorties étant mémorisé par les bascules RS, il se conçoit qu'un SET sur la sortie 3 se manifestera tant qu'il n'aura pas été annulé par un RESET sur cette même sortie (ou bien coupure intempestive d'alimentation). Prévoir des piles ou une alim. secteur, 6 à 8 V. 0,2 A. Le régulateur 2805 RC. stabilise à 5 V. Comme nous sommes sur une carte de mesure il sera bien de limiter les charges à 470 ohms*8, dans l'intérêt du régulateur, et surtout de la précision de lecture.
Moralité : priorité aux opto-couleurs, à la limite des relais ILS/DIL. de 360 ohms feront l'affaire, en revanche, il se pourrait que le régulateur réclame un petit radiateur...

Le circuit imprimé simple face 150*100 mm, comporte 10 straps, leur repérage est facilité par un ergot cuivre sur les pastilles concernés.

En dehors des CI, le potentiomètre 2 k 10 tours, serait le seul composant particulier, sans parler du connecteur 10 broches (au pas de 2,54) pour la liaison CE 158.

Les sorties « réserve » sont les vestiges d'un proto 16 voies, l'approvisionnement en 74 LS 279 n'étant pas évident, l'alternative des 4011 prend plus de place et nous ampute de 8 voies.

A la mise sous tension VART et CAN sont coincés par leurs problèmes de communication, c'est en jouant sur l'instruction OUTSTAT que l'on obtiendra le RESET propre à débloquer la situation, comme l'état de RTS est solidaire de la valeur de OUTSTAT ; nous récupérons RTS pour l'affecter au RAZ. Notez aussi qu'il suffit de tester CTS pour savoir si ANEX est sous tension. En clair, si le 1500 retourne 4, à l'appel de la fonction INSTAT : ANEX n'est pas sous tension.

A LA RECHERCHE DU SAINT-QUARTZ

Côté CP. Nous récupérons le signal d'horloge sur l'UART de la CE-158, autant dire qu'une synchro plus précise est à éviter. Donc démontage de la CE, extraction des piles en première urgence pour éviter les CC. Puis, on enlève les deux flasques latérales (2*3 vis), et nous accédons au circuit pour y souder une 4,7 K sur la broche numéro 1, l'autre aboutissant : La broche numéro 17, du connecteur Cannon DB-25(A), affectation courant en matière de RS 232C. Ainsi pas d'errements interminables à la recherche du saint quartz.

Effectuez les contrôles de routine dès la mise en route, réglez la tension de référence vers 2,5 V. Puis lancez la routine de mesure, la mise au point est terminée quand l'indication du PC-1500 correspond à la lecture du voltmètre.

J. LOUTTER

NAVIGATION ASTRO

Devant le succès considérable remporté par l'excellent programme de navigation astronomique de Monsieur VINCENT, sur PC 1211 ; et pour répondre à de très nombreuses demandes, nous vous proposons une nouvelle version de NAVIGATION ASTRO. Ce programme résoudra les nombreux problèmes d'adaptation posés par la version 1211.

UTILISATION DU PROGRAMME NAVIGATION

REMARQUES GENERALES

• L'option « IMPRIMANTE » n'est rappelée, pour chaque séquence, que si la CE 150 est connecté. OUI (1) NON (2)

• Les formats d'introduction sont indiqués dans l'affichage, les Intercepts (Ic) sont en milles, + vers l'astre, – sens inverse, les azimuts (Z) et ascensions Verses (AVA) de 0 à 360°, les déclinaisons D 0 à 90° – si Sud. Les positions (L Latitude, G longitude) en (D. MS), les heures en (H. MS) comme les introductions.

• ENTER fait avancer le Programme

• Chaque séquence est initialisée en appuyant sur la touche située sous :

Astr	RESERVE I	(Initialisation Point Astro)
Rout	RESERVE II	(Initialisation Problèmes de routes)
Mare	RESERVE III	(Initialisation Problèmes de marées)

ASTR

• L'initialisation Astro est valable 24 h au moins.

• Noter les Ic, Z qui serviront soit au graphique soit au calcul. L'heure TU s'affiche battant la seconde. Appuyer « S » et introd. successivement : la date, l'heure TU et le Point estimé choisi pour le calcul, la Rf (Route fond), Vf (vitesse fond). On voit alors, pour information, s'afficher l'Hre TU du lever et coucher du Soleil... choisir la séquence correspondant à l'astre observé.

• La hauteur observée à introduire = Hi (Hautr. instrumentale) + ε erreur instrumentale. Pour soleil et Lune tenir compte du bord observé : Infr. (I) Supr. (2) Centre (0), cas du Soleil à l'horizon artificiel.

Il est toujours possible, pour tous les astres, de recourir aux Ephémérides du SHOM., OUI (I), NON (2). Noter que les EPHm programmées restent très précises (0'7) au pire, jusqu'en 2030.

Sol

Option II : Méridienne (cumultation). Après lecture de l'Hre TU de la méridienne, introduire la hauteur observée (Ho) à cet instant, lire la Latitude méridienne et Ic, Z correspondants.

Option I : Droite de hauteur. Introd. TU d'observation et Ho du bord observé... et lecture de Ic, Z

Eto
Introduire TU d'observation et Ho

Option I : Etoile connue. Introd. le n° de l'étoile lu sur la liste des 81 étoiles... lire et noter Ic, Z

Option II : Etoile inconnue. Introd. Z et lire AVA, D ; en déduire le NOM de l'étoile (v. liste)... la suite va à l'Option I

Pla
Introd. TU d'observation. Sélectionner la planète : Vénus : I, Mars : 2, Jupiter : 3, Saturne: 4, Introd. : Ho... lire et noter Ic, Z

Lun
Introd. TU d'observation, Ho ainsi que le bord observé... Ic, Z

Pos
Option I : Lecture du Point estimé transporté à l'heure de la dernière observation. (à com-

parer au Pt. Obs. à la même Hre)
Option II : Point observé à l'instant de la dernière observation : Introd. successivement : Ier Ic, Ier Z... 2e Ic, 2e Z : lecture d'un point. On est invité, alors, à introduire un nouveau couple Ic, Z (I), si NON (2)... A chaque introduction, le point précise tenant compte des couples précédents. (Eviter d'introduire un couple manifestement « faux »).

Le point adopté, au final, prend la place du point estimé en mémoire, en vue d'une observation Astro ultérieure : (l'introd. d'une heure d'observation provoque automatiquement le transport du point estimé présent en mémoire !)...

Si on ne veut pas transporter le point, faire Vf = 0 lors de l'initialisation.
('')

Avant Rf, Vf on introduit l'élévation de l'œil au dessus de la mer (en mètres)

ROUT

Introd. les positions de départ et d'arrivée aux formats indiqués par l'affichage, puis sélectionner au choix :
Lox
Affiche l'angle de route (R) et la distance (D) loxodromique en milles entre les deux points.
Vtx
Affiche route (R) et distance orthodromique (D) en milles ainsi que les latitudes (L), longitude (G) du Vertex.

Introduire alors, le « pas » en longitude pour obtenir les latitudes correspondantes des points sur l'Ortho. Le programme affiche ensuite la distance totale Loxodromique de la route passant par les points qui jalonnent ainsi l'ORTHO.
Lim
Introduire la latitude limite (celle qu'on ne veut pas dépasser). Dans ce cas, la longitude du Vertex se situe entre celles de départ et d'arrivée. Introduire le « pas » en longitude pour jalonner les points sur les tronçons d'Ortho de départ et d'arrivée. Le programme affiche alors la distance totale loxo entre les points ainsi définis, ainsi que sur le parallèle limite.
V, C
Option I : Voile : Vent réel ? (I) : Introd. le Vent apparent bateau (VAB), route et vitesse surface (Rs, Vs), le courant pour obtenir : Vent réel (VR), Route et Vitesse fond (Rf, Vf), VMG (+ gain ou – perte en milles vers le Vent réel), Gisement vrai (Gv), Vent apparent courant (VAC)
Vent apparent ? (2) : Introduire Rs, Vs, courant et VR, pour avoir : VAB, Rf, Vf, VMG, VAC
Option II : Moteur : Route fond ? : (I) Introd. Rf, Vs et le courant pour obtenir Rf, Vf.

LISTE DES ETOILES

N°	CONSTELLATIONS	NOMS	AVA. D
1	α ANDROMEDE	ALPHERATZ	358.174+28.975
2	β CASSIOPEE	CAPH	357.992+59.034
3	γ PEGASE	ALGENIB	356.969+15.067
4	α PHENIX	ANKAA	353.689–42.420
5	α CASSIOPEE	SCHEDIR	350.172+56.423
6	β BALEINE	DENEB KAITOS	349.365+18.101
7	γ CASSIOPEE	TSIH	346.143+60.602
8	β ANDROMEDE	MIRACH	342.862+35.509
9	α ERIDAN	ACHERNAR	335.766–57.343
10	α BELIER	HAMAL	328.504+23.363
11	α PETITE OURSE	POLAIRE	327.200+89.169
12	α BALEINE	MENKAR	314.703+04.009
13	β PERSEE	ALGOL	313.300+40.875
14	α PERSEE	MIRFAK	309.296+49.787
15	α TAUREAU	ALDEBARAN	291.321+16.468
16	β ORION	RIGEL	281.618–08.224
17	α COCHER LA CHEVRE		281.216+45.977
18	γ ORION	BELLATRIX	279.000+06.331
19	β TAUREAU	EL NATH	278.760+28.590
20	ε ORION	ALNITAM	276.208–01.217
21	ζ ORION	ALNITAK	275.070–01.957
22	α ORION	BETELGEUSE	271.492+07.404
23	β COCHER	MENKALINAN	270.504+44.946
24	β GRAND CHIEN	MIRZAM	264.070–17.948
25	α CARENE	CANOPUS	264.128–52.682
26	γ GEMEAUX	ALHENA	260.885+16.418
27	α GRAND CHIEN	SIRIUS	258.944–16.686
28	ε GRAND CHIEN	ADHARA	255.550–28.941
29	ζ GRAND CHIEN	WEZEN	253.110–26.362
30	α GEMEAUX	CASTOR	246.689+31.934

LISTE DES PORTS (par numéro)

N° NOMS

100 BOUCAU
102 BOULOGNE
104 BREST
106 CALAIS
107 CHERBOURG
109 DIEPPE
111 DUNKERQUE
112 FECAMP
113 Le HAVRE
115 La ROCHELLE
117 LORIENT
118 Port TUDY
120 Pte de GRAVES
121 ROSCOFF
123 St-HELIER
125 St-Jean-de-LUZ
126 St-MALO
128 St-NAZAIRE

Les lignes 100 à 199 sont réservées au stockage des constantes harmoniques (v. Doc. N°540 du SHOM)
Exemple d'introduction des données du Port de BREST :
104 : DATA 47, 20, 67, 415, 2040, 746, 19, 54, 34, 243, 294, 342, 269, 118, 138, 178, 117, 164
105 : DATA 239, 452
Pour modifier la liste des ports, faire UNLOCK pour accéder au programme. Il faut trouver 3.88 mètres de hauteur d'eau le 1er 01 1982 à 0 h 00 (Hre en usage)

L'AVA (ascension Verse) et D (déclinaison) sont en degrés décimaux, servent à la reconnaissance. Le numérotage (1 à 81) est celui des EPHém. marines.

Route surface ? (2) : Introd. Rf, Vs et le courant pour avoir : Rs, Vf.
Est
Option I : Entretien de l'estime (I) (à faire en cours de journée en cas de changement de route et/ou de vitesse) Introduire l'Hre du changement de route(h Tu) ou d'arrivée, la nouvelle Rf, Vf et lire le Point par la séquence « Pos » en RESERVE I
Option II : Autre problème d'estime ? (2) : Introduire l'heure e la Position de départ, Rf, Vf , puis l'heure du changement de route (ou d'arrivée), nouvelle Rf, Vf et lire le Point par la séquence « Pos » en RESERVE I
La séquence V, C est indépendante et doit s'utiliser seule.
Les séquences Est et Pos peuvent s'utiliser avec Astr

MARE

Introduire la date au format indiqué, le décalage horaire : en France : 0 en hiver, I en été, le numéro du port (voir la liste). Pendant le calcul préliminaire, la phase de la Lune s'affiche ! Sélectionner alors au choix une des séquences :
Hr
Introduire l'heure en usage à laquelle on veut obtenir cette hauteur... et lire la hauteur d'eau.
Hre
Introduire la hauteur d'eau, puis l'heure approximative vers laquelle se situe l'heure correspondant à la hauteur introduite... et lire cette heure.
P, B
Affichage de la liste des Pleines mer, basses mer de la journée (heures et hauteurs)... après environ 1 minute de calculs !
Le document n° 40 du SHOM contient les Constantes harmoniques de 855 PORTS du monde. Le programme reste valide que les marées soient du type diurne, semi-diurne ou mixte.

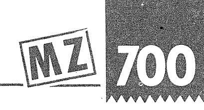
Les coordonnées moyennes des étoiles (dépouillées des nutations, aberrations) ont été calculées pour l'écliptique de 1979, 0

LISTE DES ETOILES

N°	CONSTELLATIONS	NOMS	AVA. D
31	α PETIT CHIEN	PROCYON	245.450+05.279
32	β GEMEAUX	POLLUX	243.992+28.078
33	ζ POUPE		239.282−39.945
34	γ VOILES		237.780−47.273
35	ε CARENE	AVIAR	234.480−59.441
36	δ VOILES		228.970−54.629
37	λ VOILES	SUHAIL	223.187−43.347
38	β CARENE	MIAPLACIDUS	221.758−69.629
39	ι CARENE		220.857−59.185
40	α HYDRE Femelle	ALPHARD	218.362−08.566
41	α LION	REGULUS	208.187+12.070
42	β GRANDE OURSE	MERAK	194.860+56.495
43	α GRANDE OURSE	DUBHE	194.391+61.860
44	δ LION	ZOSMA	191.753+20.639
45	β LION	DENEBOLA	183.005+14.690
46	γ GRANDE OURSE	PHECDA	181.818+53.812
47	γ CORBEAU	GIENAH	176.318−17.423
48	α CROIX	ACRUX	173.657−62.981
49	γ CROIX	GACRUX	172.504−56.994
50	β CENTAURE		169.910−48.840
51	β CROIX	MIMOSA	168.380−59.573
52	ε GRANDE OURSE	ALIOTH	166.725+56.074
53	ζ GRANDE OURSE	MISAR	159.234+55.036
54	α VIERGE	L'EPI	158.980−11.051
55	ζ GRANDE OURSE	ALKAID	153.324+49.418
56	β CENTAURE	AGENA	149.419−60.272
57	θ CENTAURE	MENKENT	148.641−36.267
58	α BOUVIER	ACTURUS	146.326+19.290
59	α CENTAURE	RIGIL KENT	140.460−60.749
60	β PETITE OURSE	KOCHAB	137.317+74.241
61	α Cne. BOREALE	LA PERLE	126.554+26.785
62	α SCORPION	ANTARES	112.972−26.386
63	α TRIANGLE AUSTRAL	ATRIA	108.393−68.991
64	ε SCORPION		107.807+34.253
65	λ SCORPION	SCHAULA	096.955−37.089
66	α OPHIUSCUS	RASALHAGUE	096.511+12.574
67	γ DRAGON	ETAMIN	090.971+51.493
68	ε SAGITTAIRE		084.306−34.395
69	α LYRE	VEGA	080.943+38.763
70	σ SAGITTAIRE	NUNKI	076.517−26.322
71	α AIGLE	ALTAIR	062.561+08.813
72	γ CYGNE		054.634+40.189
73	α PAON	PEACOCK	054.001−56.803
74	α CYGNE	DENEB	049.821+45.205
75	α CEPHEE	ALDERAMIN	040.484+62.498
76	ε PEGASE	ENIF	034.216+09.779
77	α GRUE	ALNAIR	028.271−47.064
78	β GRUE		019.646−47.000
79	α POISSON AUST.	FOMALHAUT	015.877−29.734
80	β PEGASE	SCHEAT	014.308+27.970
81	α PEGASE	MARKAB	014.071+15.092




```
  1 USING "####.##":RETURN
  2 USING "####.####":RETURN
  3 USING "###.##":RETURN
  4 USING "#####.#":RETURN
  5 FOR NU=1TO ND:READ W,U:X=X+U*COS W,Y=Y+U*SIN W:NEXT NU:RX=SQR (Y
^2+X^2 )+.1:FI=ACS (X/RX )*SGN Y+180
  6 RX=INT FI+RX/1E4,X=0,Y=0:RETURN
  7 IF IM=1LPRINT :RETURN
  8 CLS :WAIT :PRINT "RCL ?":GOTO 8
  9 HL=481268*T-89.6,PL=4069*T-25.7
 10 NL=1934*T-259,MS=36000.769*T-80.309,Z=2*MS:RETURN
 11 GOSUB 95:GOSUB 96:INPUT "Decalage Hre(H.M )?:";D,"No du Port?:";A
:DD=DEG D:GOSUB 770
 12 IF IM=1LPRINT :LPRINT "DATE:";DE:LPRINT "Decalage Hre:";DD:LPRIN
T "Port No:";A
 13 GOSUB 9:RESTORE A:FOR X=1TO 19:READ @(X ):NEXT X:READ TE,U
 14 GOSUB 6:DT=MS-2*HL-90:RESTORE 560
 15 ND=6:GOSUB 5:R1=RX,ND=11,DT=Z-2*HL:GOSUB 5:R2=RX,ND=3:GOSUB 5:R4
=RX
 16 IF HD=1LET DT=0,HD=A*COS (MS-K ):BEEP 5:GOTO 8
 17 HD=1,T=T+1/36525:GOSUB 9:R7=R1,R8=R2,R9=R4:GOTO 14
 18 GOSUB 4:LPRINT "R";A;",D";S:RETURN
 19 GOSUB 3:PRINT ZE$;DMS H;" TU":GOSUB 2:PRINT "L";DMS W;",G";DMS Z
:RETURN
 20 H=HD:G=179
 21 POKE 1099,G:RESTORE 27:FOR G=12TO 14:GOSUB 22:@(G )=R,@(G+3 )=F:NE
XT G:G=L*SIN O+2*M*SIN P+4*N*SIN Q:RETURN
 22 READ A,B,J:F=B-A:IF F>180LET F=F-360
 23 F=A-180+T/24*(J*360+F ),R=A-INT A+T/24*(B-A+INT A-INT B ):WAIT
 24 H=H+1E4*R*COS F,X=U/100+H/1E3:RETURN
 26 WAIT :PRINT USING "#####.#";"R";A;",D";S:RETURN
 27 DATA R7,R1,1,R8,R2,2,R9,R4,4
 28 GOSUB 3:LPRINT ZE$;DMS H;"h":GOSUB 2:LPRINT "Lat";DMS L:LPRINT "
Lon";DMS G:USING :LPRINT "Rf";R;",Vf";F:RETURN
 29 G=T-U:GOSUB 64
 30 L=Q:GOSUB 61:L1=LC,L=I:GOSUB 61:D=L1-LC,Y=G*60:GOSUB 62:GOSUB 4:
S=S+ABS ((Q-I )*60*M/D ):RETURN
 32 INPUT "Hre locale (H.M )?:";T:T=DEG T-DD:GOSUB 20
 33 AX$="v":BEEP 5:IF G<0LET AX$="^"
 34 GOSUB 3:T=DMS (T+DD ):PRINT "a";T;"h ";AX$;" Hr";X:GOSUB 7:LPRINT
 AX$;T;"h  Hr";X:GOTO 8
 35 D=B,F=C:GOSUB 71:RETURN
 36 GOSUB 760
 37 H=DEG H,I=H-H2,H2=H,T=T+I/876600,C=C+I*15.041,D=I*F/60
 38 G=G-D*SIN R/COS (L+D/2*COS R ):L=L+D*COS R:RETURN
 39 GOSUB 43:P=.2665/I:RETURN
 40 GOSUB 10:M=35999.05*T-1.524,K=SIN M*(1.919+K1*T )+SIN (2*M )*.02
 41 S=MS+K+NG,I=9997/(1E4+167*COS (M+K )):RETURN
 42 H=DMS (12+(G+B )/15 ):GOSUB 37:GOSUB 43:B=ASN SIN (N-C+180+H*15 ),H
=DMS H:RETURN
 43 GOSUB 40:J=S,K=0:GOSUB 98:RETURN
 45 INPUT "EPhm.?:1,si NON:2:";X:RETURN
 46 INPUT "Paral.?(M.m ):";Q:Q=Q*50/3:RETURN
 47 INPUT "AVa?(D.MS ):";N:N=360-DEG N:GOTO 50
 48 INPUT "1/2diam?(M.m ):";P:P=P/60:RETURN
 49 INPUT "AHaO?(D.MS ):";J:J=DEG J-G
 50 INPUT "Decl?(D.MS-si SUD ):";O:O=DEG O:RETURN
 51 BEEP 5:INPUT "Hr.Obs?(D.MS ):";HO:V=DEG HO:V=V-.0293*SQR E-.0012-
.0154/TAN V+Q/1E3*COS V
 52 IF P<>0INPUT "Bord SuP:2,Inf:1,Centr:0?";D:IF D=2LET P=-P
 53 V=V+P*SGN E:RETURN
 56 PRINT ZE$:IF IM=1LPRINT ZE$:LPRINT "Pas";P
 57 GOSUB 74:RETURN
 58 AX$="hBM":IF V=1LET AX$="hPM"
 59 X=DMS (X+DD ):V=-V:PRINT AX$;X;",Hr";Y:IF IM=1LPRINT AX$;X;",Hr";
 60 RETURN
 61 LC=7915.6*LOG (TAN (45+(L-0.197*SIN (2*L ))/2 )):RETURN
 62 M=SQR (Y^2+D^2 ):A=ACS (D/M )*SGN Y-180*NOT SGN -Y:RETURN
 64 G=ACS COS G*SGN SIN G:RETURN
 66 INPUT "Pas en Long(DEG )?:";P
 67 GOSUB 64:NA=INT ABS (G/P ):RETURN
 68 A=.3478*I,B=.0039*I:IF T<1.07LET A=.2606*I;B=.0044*I
 69 RETURN
 70 WAIT 0:GOSUB 3:PRINT "DePuis";DMS H;"h:";:INPUT "Rte Fnd?";R:CLS
:INPUT "Vf ?:";F:WAIT :RETURN
 71 W=ACS (TAN O/TAN J ),G=F+Z*W:GOSUB 64:GOSUB 1:RETURN
 72 WAIT :GOSUB 1:PRINT "L";DMS Q;",G";DMS G:RETURN
 73 LPRINT "L";DMS Q;",G";DMS G:RETURN
 74 FOR R=1TO NA:Q=ATN (TAN J*COS (R*P-W )):G=T+Z*P:GOSUB 64:U=G:GOSU
B 72:IF IM=1GOSUB 73
 75 GOSUB 29:T=U,I=Q:NEXT R:RETURN
 78 J=C-G-N,W=1
 79 X=COS J*COS O*COS L+SIN L*SIN O
 80 I=ASN X,Y=SIN J,D=W*(COS J*SIN L-TAN O*COS L ):GOSUB 62:RETURN
 81 U=H,Q=E,T=C,I=B:RETURN
 82 GOSUB 205:INPUT "Point Estim:1,Obser:2 ?:";X:IF X=1LET W=L,Z=G:G
OSUB 19:GOTO 89
 83 M=0,N=0,P=0,ZE$="Pt.Obser a:"
 84 P=P+1:INPUT "1er Intc?:";A,"Z?:";J,"2eme Intc?:";B,"Z?:";D
 85 O=(B*COS J-A*COS D )/(COS J*SIN D-COS D*SIN J )/60,N=N+O/COS L,M=M
-TAN J*O+A/COS J/60:GOSUB 3
 86 W=L+M/P,Z=G-N/P:GOSUB 19
 87 INPUT "Point Suivt?:1,si NON:2:";X:IF X=1GOTO 84
 88 L=W,G=Z
 89 GOSUB 7:GOSUB 28:GOTO 8
 90 GOSUB 78:GOSUB 51
 91 A=A+180*SGN Y:GOSUB 1:I=(V-I )*60:PRINT "Ic=";I;",Z=";A
 92 GOSUB 7:GOSUB 2:LPRINT "TU Obs :";DMS H:LPRINT "Hr Obs:";HO:GOSU
B 1:LPRINT "Ic";I;",Z";A:GOTO 8
 93 GOSUB 95:WAIT 0:PRINT "DePart.:";:GOSUB 200:IF IM=1LPRINT :GOSUB
 28
 94 GOSUB 205:INPUT "h TU Chgt.Rte?(ou Arriv ):";H:GOSUB 37:GOSUB 70:
GOTO 89
 95 LOCK :P=100:GOSUB 740:RETURN
 96 INPUT "DATE(JM.A )?:";D:J=INT (D/P ),M=INT (D-J*P ),Y=(D-INT D )*P^2
:IF M<=2LET Y=Y-1,M=M+12
 97 A=INT (Y/P ),T=(INT (365.25*Y )+INT (30.6*(M+1 ))-A+INT (A/4 )+J-694
023.5 )/36525:DE=D:RETURN
 98 Y=SIN J*.9175-TAN K*.3978,D=COS J:GOSUB 62:N=A,O=ASN (SIN K*.917
5+COS K*.3978*SIN J ):RETURN
 99 INPUT "Entretien Est:1,Autre:2 ?";X:ON XGOTO 94,93
100 DATA 70,20,66,57,246,1189,410,10,34,39,283,288,339,87,107,128,16
1,172,176,214
101 DATA 255
102 DATA 0,19,31,50,536,2895,929,119,340,213,0,86,115,138,337,358,52
,253,277,333
103 DATA 490
104 DATA 47,20,67,65,415,2040,746,19,54,34,243,294,342,269,118,138,1
78,117,164
105 DATA 239,452
106 DATA 81,25,50,26,441,2459,779,90,254,150,219,100,160,81,352,15,6
9,283,309,6,400
107 DATA 61,19,59,91,368,1870,699,47,138,84,243,330,13,120,238,257,3
01,24,50,100
108 DATA 383
109 DATA 41,13,46,96,575,3126,1031,96,270,185,282,31,49,128,318,340,
31,218,243
110 DATA 298,484
111 DATA 70,32,86,33,372,2116,627,57,156,87,233,111,177,12,357,22,75
,310,337,35,324
112 DATA 94,8,43,86,493,2679,870,0,125,81,153,342,38,139,306,326,14,
0,173,226,456
113 DATA 75,11,49,94,491,2608,878,89,252,174,236,357,32,131,294,315,
3,113,136,190
114 DATA 487
115 DATA 58,24,73,62,362,1743,631,107,248,106,230,301,342,89,109,129
,162,25,72
116 DATA 159,384
117 DATA 0,22,65,60,316,1500,546,41,80,22,0,301,345,92,110,130,163,3
7,87,183,300
118 DATA 193,19,61,64,307,1471,546,35,77,24,231,290,343,90,105,125,1
58,33,83,183
119 DATA 309
120 DATA 49,25,71,45,306,1575,521,31,75,21,255,304,346,99,119,138,17
3,18,69,176,321
121 DATA 28,22,71,87,534,2689,1015,41,105,65,242,295,345,100,153,171
,218,168,200
122 DATA 255,519
123 DATA 37,29,81,90,646,3360,1324,71,199,147,223,233,346,97,166,182
,231,279,300
124 DATA 355,606
125 DATA 50,29,71,61,280,1336,463,13,29,13,94,290,333,80,100,121,154
,341,31,107,250
126 DATA 54,24,88,90,720,3680,1432,101,273,206,224,330,354,126,190,2
08,258,315
127 DATA 337,35,671
128 DATA 89,23,66,58,358,1710,605,75,180,70,241,298,347,93,116,136,1
70,57,108,215
129 DATA 320
130 INPUT "Rf?:1,Rv?:2:";X:IF X=2GOTO 135
132 GOSUB 160:GOSUB 172:GOTO 8
135 INPUT "Rf?:";R:GOSUB 162:H=R-J,V=ASN (K/B*SIN H ):Z=R+V
137 GOSUB 182:PRINT "Rv:";A;",Vf:";K*COS H+B*COS V:GOTO 8
155 INPUT "Voile?:1,Moteur:2:";X:USING "####.#":IF X=2GOTO 130
157 INPUT "Ve Reel?:1,APP.:2:";X:ON XGOTO 165,175
159 INPUT "GA(0 a 360 )?:";S,"VAB?:";U
160 INPUT "Rs?:";I
162 INPUT "Vs?:";B,"Ct(Nds )?:";K,"Ct(0 a 360 )?:";J:FOR Y=2TO 21:@(Y )
=@(Y )+.01:NEXT Y:RETURN
165 GOSUB 159:D=B-U*COS S,Y=U*SIN S:GOSUB 62:Q=180-A,W=Y/SIN Q:X=I+Q
-J
166 D=K-W*COS X,Y=W*SIN X:GOSUB 62:N=Y/SIN A,O=I+Q+ASN (K/N*SIN X ):Z
=O:GOSUB 182
167 PRINT "VR:";N;"nd,du";A:GOSUB 172
170 PRINT "VMG:";M*COS (O-A );",GV";Q:Z=I+Q:GOSUB 182:PRINT "VAC du";
A;W;"nd":GOTO 8
172 D=B*COS I+K*COS J,Y=B*SIN I+K*SIN J:GOSUB 62:PRINT "Vf:";M;",Rf:
";A:RETURN
175 INPUT "VR(0 a 360 )?:";O,"(Nds )?:";N:GOSUB 160
180 X=180+O-J:Y=N*SIN X,D=K-N*COS X:GOSUB 62:P=M,Z=O+ASN (K/P*SIN X )
:GOSUB 182:Q=A-I:Y=P*SIN Q
181 D=B+P*COS Q:GOSUB 62:PRINT "VAB:";M;"nd";",GA";A:GOSUB 172:GOTO
170
182 Y=SIN Z,D=COS Z:GOSUB 62:RETURN
200 INPUT "h TU Est(H.M )?:";H:GOSUB 37:CLS :INPUT "Esti:Lat(D.M )-si
SUD?:";L
201 INPUT "Long-si EST?:";G:L=DEG L,G=DEG G:GOSUB 70
205 ZE$="Pt.Estim a:":RETURN
212 WAIT 0:USING "###.####":PRINT "hTU";(TIME /100-INT (TIME /100 ))*
100;" APP:S..Suite.."
214 IF INKEY$ <>"S"GOTO 212
220 WAIT :GOSUB 95:GOSUB 96:K1=-.006:GOSUB 10:NG=-K1*SIN NL,NO=.0025
*COS NL
224 C=.2769194+100.002136*T,C=(C-INT C )*360+NG*.9:H2=0:GOSUB 200
233 INPUT "el Deil(Metr )?:";E:GOSUB 750:GOSUB 7:USING :LPRINT "DATE:
";DE:LPRINT "el Deil:";E:GOTO 89
238 Q=2.5:INPUT "Droite Hr:1,Merid:2 ?";X:IF X=2GOTO 262
239 GOSUB 36:GOSUB 45:IF X=2GOSUB 39:GOTO 90
240 GOSUB 49:GOSUB 48
241 W=1:GOSUB 79:GOSUB 51:GOTO 91
262 B=0:GOSUB 42:GOSUB 42:GOSUB 2:PRINT "TU Mer:";H;",Lat..":GOSUB 4
5:IF X=1GOSUB 50:GOSUB 48:GOTO 269
268 GOSUB 37:GOSUB 39
269 GOSUB 51:A=DMS (O+SGN L*(90-V ))
274 PRINT "Lat.Mer:";A;" Ic,Z...":IF IM=1LPRINT :LPRINT "TU Mer:";DM
S H:LPRINT "Lat Mer";A
276 GOSUB 78:GOTO 91
291 GOSUB 36:GOSUB 45:IF X=1GOSUB 49:GOSUB 46:P=0:GOTO 241
292 INPUT "Venus:1,Ma:2,Ju:3,Sa:4 ?";A:ON AGOTO 300,308,332,340
296 DATA 58519.212*T+342.767,U-1.408*T-130.164,.9*T+75.78,.00678,.72
333,3.394,K1
300 N=0:RESTORE 296
302 READ U,P,S,X,V,B,W
304 Z=P:FOR A=0TO 6:Z=P+X*180/PI *SIN Z:NEXT A:Y=SIN (Z/2 )*SQR ((1+X
)/(1-X )),D=COS (Z/2 ):GOSUB 62
305 V=V*(1-X*COS Z ),Q=U+N+2*A-P-S,Y=COS B*SIN Q,D=COS Q:GOSUB 62:O=A
+S,N=ASN (SIN Q*SIN B ):GOSUB 40
306 S=S-K1-NG:Z=O-S,Y=SIN Z*V*COS N,D=COS Z*V*COS N+I:GOSUB 62:Y=V*S
IN N,X=SQR (M^2+Y^2 )
307 Q=2.5/X,K=ASN (Y/X ),J=A+S+K1*COS Z+W+NG:GOSUB 98:P=0:GOTO 90
308 A=16105*T,B=A*.812,D=A*.142,I=A/.955,J=A*.905:RESTORE 322:O=9
314 K=0:FOR X=1TO O:READ N,Y:K=K+N*COS Y:NEXT X:N=K/1E4:GOTO 302
322 DATA 66,217-A,66,57+B,42,20+2*A,42,100-D,21,74+I,21,J-83,21,32+1
098*T,12,47+2*D
324 DATA 12,243+3035*T,19141.696*T-66.25,U-1.84*T+25.769,.771*T+48.7
86,.093396
326 DATA 1.52369,1.85,K1
332 RESTORE 377:N=366:GOTO 345
340 RESTORE 361:N=396
345 Q=130
350 IF 100*T<QREAD A,B,D,P,Q:GOTO 350
355 N=(A*COS ((100*T-P )/D*360 )+B )/1E3:RESTORE N:GOTO 302
361 DATA -174,-44,29,114,116,-330,114.2,52,115,102,97.7,120,14,98.5,
97
362 DATA -400,-100,58,75,83.7,-295,-100,48,80,74
366 DATA 3036.302*T+238.049,U-1.61*T-12.72,1.01*T+99.443,.048482,5.2
0256,1.304,K1/2
377 DATA 32,3.9,123,123,14.2,21.7,116,116,45,17,10,114,112,33.4,10,9
,104,102,-62.5
378 DATA -18,10,99,100,-111.4,26.9,12,87.75,88,121.7,37,11,82.7,83,3
6,123,8,74.5,74
```



ECHELLE :

ECHELLE argument donne à la variable SCALE (apparaissant à l'écran sous le nom S) la valeur de argument. Son effet est de changer les déplacements relatifs du stylo.
Avec ECHELLE 2 AVANCES 50 fait avancer le stylo de 50 unités (pour l'imprimante du MZ-700)
Avec ECHELLE 2 AVANCES 50 provoquera une avance 100 unités.

TAILLE :

TAILLE argument multiplie SCALE par la valeur de l'argument.
Par exemple TAILLE 0.5 puis ETOILE dessinera une étoile deux fois plus petite.

LIGNE :

LIGNE argument est un entier de 1 à 16 permet de modifier le type de ligne tracée par le stylo (Voir page 87 manuel de l'ordinateur).

SOIT :

SOIT permet au programme d'utiliser des variables. SOIT COTE 15 + 7 donnera à la variable COTE la valeur 22. On remarquera que COTE est alors une variable considérée comme celles du programme, par le Basic. Autrement dit, il ne faudra pas utiliser comme variable une des variables du programme lui même, sous peine de le voir fonctionner de façon aberrante.

SI. SINON :

C'est le IF THEN ELSE du Basic : SI condition action I SINON action 2. Il peut être utile pour la condition, d'utiliser certaines fonctions du Basic ce qui peut poser des problèmes aux programmes (pour lequel, un espace est un séparateur entre diverses commandes). On pourra alors écrire la condition entre crochets : (cf ligne 3).

```
POUR DOUBLEPOLY
Parametres: REP C0 C1 A1 F1
 1  SOIT KA A
 2  REPETES REP
 3  SI [(KA MODULO(F1*A1))
<(F1*A1/2)] AVA
NCES C0 SINON AVANCES C1
 4  DROITE A1
 5  SOIT KA KA+A1
 6  ENCORE
 7  FIN
```

UTILISATION DE PARAMETRES :

L'exemple précédent montre qu'il est possible de définir de nouvelles commandes possédant des paramètres. Il peut y avoir au maximum 5 paramètres (exemple : On entre POUR DOUBLEPOLY REP CO CI AI FI) et lors de l'appel on indique les valeurs : DOUBLEPOLY 50 50 100 30 120. Le programme affectera à REP la valeur 50, à CO la valeur 50,..., à FI la valeur 120. A la fin, REP, CO,..., FI retrouveront leurs valeurs intiales. Il y a donc, comme en Basic, passage de paramètres, par valeur.

```
POUR CARRE
Parametres: COTE
 1  REPETES 4
 2  AVANCES COTE
 3  DROITE 90
 4  ENCORE
 5  FIN

POUR CARRES

 1  SOIT COTE 0
 2  REPETES 20
 3  SOIT COTE COTE+5
 4  CARRE COTE
 5  ENCORE
 6  FIN
```

REMARQUES :

Shift F5 permet de réinitialiser l'imprimante (en cas d'erreur).
Shift F4 permet de faire redémarrer le programme après un Break, sans perdre le contenu de la mémoire.
Les valeurs X et Y affichées par le programme, en haut à gauche de l'écran sont les parties entières de X et Y, positions du stylo. Elles sont donc « arrondies » par rapport aux vraies valeurs. D'autre part, il se produira des erreurs sur X et Y car les variables sont calculées par des Sinus et Cosinus d'angles exprimés en degrés.

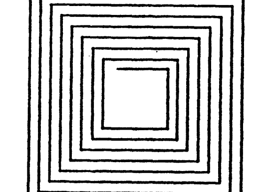
QUELQUES EXEMPLES

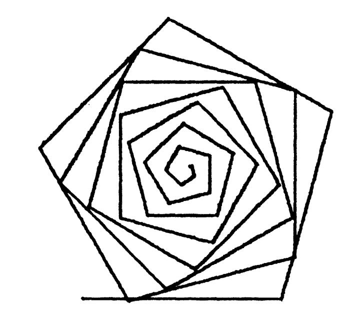
Autre définition d'un carré. Il faudra faire Shift Break pour arrêter le programme et le reprendre par Shift F4

```
POUR CARRE

 1  AVANCES 100
 2  DROITE 90
 3  CARRE
 4  FIN
```

```
POUR MAZE
Parametres: DISTANCE
 1  AVANCES DISTANCE
 2  DROITE 90
 3  MAZE DISTANCE+5
 4  FIN
```





```
POUR SPIRAL
Parametres: DISTANCE
 1  AVANCES DISTANCE
 2  DROITE 75
 3  SPIRAL DISTANCE+5
 4  FIN
```

```
  396 DATA 1223.51*T+266.564,U-1.958*T-91.098,.873*T+112.79,.05558,9.5
5475,2.49,K1/3
  403 I=100*T-79:GOSUB 45:IF X=1GOSUB 47:GOTO 276
  406 INPUT "Numero de l ETOILE ?:";X:RESTORE (1000+X):READ X,O:N=360-
X
  412 M=0:W=SIN N,Y=COS N:Z=K1*COS S*.92:IF X=3GOSUB 68:GOTO 426
  418 IF X=19LET M=K1/2
  422 A=I*(.0128-K1*W*TAN O+M),B=-I*K1*Y
  426 A=A+NG*(.92+.4*W*TAN O)+NO*Y*TAN O+(Z*Y+W*K1*SIN S)/COS O
  430 B=B+NG*.4*Y+NO*W+Z*.43*COS O-Z*W*SIN O+K1*Y*SIN S*SIN O,N=N+A,O=
O+B:GOTO 276
  436 GOSUB 36:P=0,Q=0:GOSUB 51:S=36E3*T-80:INPUT "Et.Connue:1,si NON:
2 ?";X:IF X=1GOTO 403
  439 INPUT "Azimut?:";J:O=V,W=-1:GOSUB 79:B=360-A+G-C:Y=SIN B,D=COS B
:GOSUB 62:GOSUB 1
  440 PRINT "Eti:AVa:";A;",D:";I:GOTO 403
  442 GOSUB 36:GOSUB 45:IF X=1GOSUB 46:GOTO 240
  444 GOSUB 40:B=477198.85*T-63.89,U=2*M,D=445267.11*T-9.26,J=2*D,N=2*
B
  445 I=11.25+T*483202.025+.025*SIN NL,K=2*I,O=J-B,Q=J-I,A=44:RESTORE
452
  446 W=0:FOR V=1TO A:READ X,Y:W=W+X*SIN Y:NEXT V:IF A=44LET P=W,A=32:
GOTO 446
  448 Q=951+52*COS B+9*COS O+8*COS J+3*COS N,J=(P-2*SIN NL-89563+48126
7883*T)/1E3
  450 K=W/1E3:GOSUB 98:P=272E-6*Q:GOTO 90
  452 DATA 6289,B,1274,O,658,J,214,N,-186,M,-114,K,59,J-N,57,O-M,53,J+
B,46,J-M,41,B-M
  456 DATA -35,D,-30,M+B,15,J-K,-13,K+B,11,B-K,11,O+J,10,3*B,9,2*O,-8,
O+M,-7,J+M,5
  457 DATA B-D,5,M+D,4,J-M+B,4,N+J,4,2*J,4,O-N,3,N-M,-3,O+K,2,J-M-N,-2
,B+D,2,J-U,-2
  458 DATA N+M,-2,U,2,O-U,2,K-J-B,-2,K+J,1,J-M-B,-1,N+K,-1,D+O,-1,J+B+
M,1,2*O-M,1,B-U
  460 DATA -1,O+U,5128,I,281,B+I,278,B-I,173,Q,55,O+I,46,Q-B,33,J+I,17
,N+I,1,Q+J,9
  462 DATA Q+B,9,N-I,8,Q-M,4,Q-N,4,J+I+B,-3,Q+M,2,O+I-M,2,J+I-M,2,O-I-
M,2,I-M+B,2,O+Q
  464 DATA -2,I+M,-2,3*I,2,B-I-M,-2,I+D,-2,I+M+B,1,I-M-B,1,I-M,1
  467 DATA I-D,1,I+3*B,1,O+J+I,1,B-3*I,1,I+2*O
  470 INPUT "Quelle Hr?:";TE,"Hre APProx?:";T:T=T-DO:GOTO 476
  472 IF ABS Y>6.3LET Y=Y/2:GOTO 472
  474 T=T+Y
  476 GOSUB 20:Y=(X-TE)/(5*G):IF ABS (X-TE)>.02GOTO 472
  478 GOTO 33
  480 WAIT 0:PRINT "Hre,Hr:PM,BM..":IF DT=0LET DT=1:V=0:GOTO 489
  486 GOTO 513
  489 C=-.77,I=540
  492 IF ABS V<6GOTO 495
  493 V=0,C=C+2.1
  495 T=C+V:G=&99:GOSUB 21
  498 Y=-12*G/(PI *(L*COS O+4*M*COS P+16*N*COS Q)):IF ABS Y>.1LET V=V+
Y:GOTO 492
  504 T=T+Y:IF T<24GOSUB 20:GOTO I
  507 GOTO 513
  510 I=I+3:IF I<=549GOTO 493
  513 V=W:GOSUB 3:BEEP 5:WAIT :RESTORE 515:FOR C=3TO I-540STEP 3:READ
X,Y:GOSUB 58:NEXT C:GOTO 8
  515 DATA D,Z,E,MS,S,NL,FI,RX
  540 D=T,Z=X:GOTO 510
  543 E=T,MS=X,K=D,W=-1:IF Z>XLET W=1
  544 GOTO 555
  546 S=T,NL=X,K=E:GOTO 555
  549 FI=T,RX=X,K=S
  555 IF ABS (T-K)<.3GOTO 493
  558 GOTO 510
  560 DATA MS+90-N,D,DT-M,C,DT-HL+PL-L,B,90-MS-N,-D/3,DT-NL-M,C/5.3,MS
+NL+90-N,D/7.4
  561 DATA DT-P,F,DT-HL+PL-O,E,2*DT-Z+2*PL-O,E/7.6,-Q,G,2*DT-O,E/6.3
  562 DATA DT+Z-HL-PL-O,E/5.3,Z-HL-PL-P,-F/35,Z-Q,G/3.7,283-MS-Q,G/17,
DT-NL-P,-F/27
  563 DATA Z+NL-Q,G/12,2*DT-HL+PL-R,H,2*DT-S,I,DT-TE,J
  643 GOSUB 95:INPUT "DePt:Lat(D.M)-si S ?:";B,"Lon-si E ?:";C,"Arriv:
Lat?:";E,"Lon?:";H
  644 IF IM=2GOTO 647
  646 GOSUB 1:LPRINT "DePart.:":LPRINT "L";B;",G";C:LPRINT "Arrivee:":
LPRINT "L";E;",G";H
  647 ZB$="D.Lox TOTAL",ZD$="Rte Dist,LOXO:":B=DEG B+.002,C=DEG C,E=DE
G E,H=DEG H+.001:GOTO 8
  656 S=0:GOSUB 81:WAIT 0:PRINT ZD$:GOSUB 29:GOSUB 26:GOSUB 7:LPRINT Z
D$:GOSUB 18:GOTO 8
  700 ZE$="RteDist ORT,Vtx":WAIT 0:PRINT ZE$:G=C-H:GOSUB 64:J=G,W=-1,O
=E,L=B:GOSUB 79:S=ACS X*60
  706 Z=-SGN G:GOSUB 26:V=ACS (ABS (SIN A*COS B))*SGN COS A,J=V:GOSUB
35:GX=W:Q=V:GOSUB 72
  708 IF IM=1LPRINT :LPRINT ZE$:GOSUB 18:GOSUB 73
  715 S=0,G=C-H:GOSUB 66:GOSUB 81:W=GX:J=V:ZE$="Pts. Par VERTEX:":GOSU
B 56
  717 U=H,Q=E:GOSUB 29:PRINT ZB$;S:GOSUB 7:LF -1:LPRINT ZB$;S:GOTO 8
  725 S=0:INPUT "Lat LIMIT ?:";J:J=DEG J:GOSUB 35:K=G,V1=W,O=E,F=H:Z=-
Z:GOSUB 71
  726 V=G,Z=-Z:ZE$="Long sur L.Limit":PRINT ZE$:PRINT "G1";DMS K;",G2"
;DMS V
  727 IF IM=1LPRINT :LPRINT "L:LIMITE:";DMS J:LPRINT ZE$;": G";DMS K;"
,G";DMS V
  730 ZE$="Pts. Par L.Limite",G=K-C:GOSUB 66:I=B,T=C:W=V1:GOSUB 56
  732 U=K,Q=J:GOSUB 29:G=V-K:GOSUB 64:I=Q+.002:GOSUB 30:G=H-V:GOSUB 67
:I=J,T=V,W=0:GOSUB 74:GOTO 717
  740 IF CHR$ PEEK &B002<>"M"LET IM=2:RETURN
  745 INPUT "ImPrimer?:1,si NON:2:";IM:RETURN
  750 WAIT 0:PRINT "h TU Lever Coucher SOL...":B=0:GOSUB 42
  751 IF ABS O>=90-ABS LWAIT :PRINT "ni Lever ni Coucher..!":RETURN
  752 W=(12+(G+B)/15),P=(ACS (-TAN L*TAN O))/15+.06/(SQR ((COS O)^2-(S
IN L)^2)):GOSUB 3:WAIT
  754 PRINT "hTU Lev:";DMS (W-P);" Cou";DMS (W+P):RETURN
  760 INPUT "hTU Obs(H.MS)ou:T(TIME)?:";A$:IF A$="T"LET H=(TIME /100-I
NT (TIME /100))*100:RETURN
  762 H=VAL A$:RETURN
  770 K=T*100*12.3685:M=K-INT K:D=.25/2:A$="Nouvelle LUNE"
  772 IF M>DAND M<.25+DLET A$="1er Quartier"
  774 IF M>D+.25AND M<.5+DLET A$="Pleine LUNE"
  776 IF M>.5+DAND M<.75+DLET A$="Dernier Quart."
  778 WAIT 0:PRINT "Phase LUNE: ";A$:RETURN
  1001 DATA 358.174,28.975
  1002 DATA 357.992,59.034
  1003 DATA 356.969,15.067
  1004 DATA 353.689,-42.42
  1005 DATA 350.172,56.423
  1006 DATA 349.365,18.101
  1007 DATA 346.143,60.602
  1008 DATA 342.862,35.509
  1009 DATA 335.766,-57.343
  1010 DATA 328.504,23.363
  1011 DATA 327.2,89.169
  1012 DATA 314.703,4.009
  1013 DATA 313.3,40.875
  1014 DATA 309.296,49.787
  1015 DATA 291.321,16.468
  1016 DATA 281.618,-8.224
  1017 DATA 281.216,45.977
  1018 DATA 279,6.331
  1019 DATA 278.76,28.59
  1020 DATA 276.208,-1.217
  1021 DATA 275.07,-1.957
  1022 DATA 271.492,7.404
  1023 DATA 270.504,44.946
  1024 DATA 264.07,-17.948
  1025 DATA 264.128,-52.682
  1026 DATA 260.885,16.418
  1027 DATA 258.944,-16.686
  1028 DATA 255.55,-28.941
  1029 DATA 253.11,-26.362
  1030 DATA 246.689,31.934
  1031 DATA 245.45,5.279
  1032 DATA 243.992,28.078
  1033 DATA 239.282,-39.945
  1034 DATA 237.78,-47.273
  1035 DATA 234.48,-59.441
  1036 DATA 228.97,-54.629
  1037 DATA 223.187,-43.347
  1038 DATA 221.758,-69.629
  1039 DATA 220.857,-59.185
  1040 DATA 218.362,-8.566
  1041 DATA 208.187,12.07
  1042 DATA 194.86,56.495
  1043 DATA 194.391,61.86
  1044 DATA 191.753,20.639
  1045 DATA 183.005,14.69
  1046 DATA 181.818,53.812
  1047 DATA 176.318,-17.423
  1048 DATA 173.657,-62.981
  1049 DATA 172.504,-56.994
  1050 DATA 169.91,-48.84
  1051 DATA 168.38,-59.573
  1052 DATA 166.725,56.074
  1053 DATA 159.234,55.036
  1054 DATA 158.98,-11.051
  1055 DATA 153.324,49.418
  1056 DATA 149.419,-60.272
  1057 DATA 148.641,-36.267
  1058 DATA 146.326,19.29
  1059 DATA 140.46,-60.749
  1060 DATA 137.317,74.241
  1061 DATA 126.554,26.785
  1062 DATA 112.972,-26.386
  1063 DATA 108.393,-68.991
  1064 DATA 107.807,-34.253
  1065 DATA 96.955,-37.089
  1066 DATA 96.511,12.574
  1067 DATA 90.971,51.493
  1068 DATA 84.306,-34.395
  1069 DATA 80.943,38.763
  1070 DATA 76.517,-26.322
  1071 DATA 62.561,8.813
  1072 DATA 54.634,40.189
  1073 DATA 54.001,-56.803
  1074 DATA 49.821,45.205
  1075 DATA 40.484,62.498
  1076 DATA 34.216,9.779
  1077 DATA 28.271,-47.064
  1078 DATA 19.646,-47
  1079 DATA 15.877,-29.734
  1080 DATA 14.308,27.97
  1081 DATA 14.071,15.092
 65279 END
```





Essayez ces 2 programmes :

10 TIME=0:FOR I=-5000TO 0:NEXT I:LPRINT TIME
20 TIME=0:FOR I=-5000TO 1:NEXT I:LPRINT TIME
RUN
0.0057
0.0106

Bizarrement, le PC 1500 gagne 9 secondes en effectuant 1 boucle de plus. La raison vient du fait que le Basic possède un test spécial pour une fin de boucle ayant pour valeur 0 ; d'où gain assez considérable de temps.

P. MATSIS



Dans un programme Basic où la rapidité d'exécution est un facteur important, il est préférable de séparer 2 assignations consécutives par une virgule au lieu de 2 points. L'interpréteur assimilera alors cette double assignation à une seule instruction et l'exécutera 2 fois plus vite.

(3 fois s'il y a 3 assignations).
```
10 TIME=0
20 FOR I=1TO 500
30 A=1;B=1;A$="PC 1500"
40 NEXT I: PRINT TIME
```
Ce exemple de programme est exécuté 3 fois plus vite que le même programme muni de 2 ":" en ligne 30.

E. BEAUREPAIRE



L'instruction AREAD du Basic possède une particularité qui n'est mentionnée dans aucun manuel : Elle peut être utilisée comme un POINT très puissant testant plusieurs colonnes simultanément (jusqu'à 40). En effet, si elle est appelée par un DEF alors qu'il y a un graphisme à l'écran, elle retourne dans A$ (si on fait AREAD A$) la forme des premières colonnes de l'écran en hexadécimal. Elle renverra l'état de 8 colonnes dans A$ puisque A$ contient 16 caractères ; mais si on dimensionne une chaîne de 80 caractères (ex : A$(0)*80), on pourra tester 40 colonnes de l'afficheur !

Exemple :
```
10 CLS:WAIT:GPRINT"010204081020407F"
20 "A"AREAD A$:PRINT A$
```
Faites RUN puis appelez AREAD par DEF A ; le PC affiche le contenu de A$, c'est-à-dire la forme graphique qui était sur les 8 premières colonnes de l'écran : 010204081020407F

La seulle limitation à cet emploi inatendu de AREAD est qu'on ne peut l'appeler autrement que par DEF. On peut cependant imaginer des applications telles qu'un stockage immédiat d'un dessin sur l'écran dans A$(0) en faisant DEF A.

E.B.



Le fait de pouvoir merger des programmes permet d'en écrire plusieurs en mémoire sans avoir à se soucier des numéros de lignes. Cette opération n'est, en temps normal, possible qu'avec un CE 150. Si vous n'en disposez pas, voici comment faire :

Soit un programme en mémoire :
```
POKE &734E,PEEK&7865,PEEK&7866
NEW STATUS 2
```
Tapez le second programme puis :
```
POKE&7865,PEEK&734E,PEEK&734F
```
Vous pouvez répéter cette opération à volonté pour merger plusieurs programmes successifs en mémoire.

M. MENU

LANGAGE MACHINE (6)

LES ROUTINES DE CALCULS

Nous allons poursuivre cette fois-ci avec l'étude des principales routines de traitement numérique ou alphabétique. Tout d'abord nous allons définir la notion de registre arithmétique, avant d'étudier une à une les routines les plus utiles.

LES REGISTRES ARITHMETIQUES :

Ces registres sont utilisés de manière interne pour le traitement des opérations, aussi bien sur des chaines que sur des nombres. Il occupent chacun huit octets et sont comparables aux registres statiques A-Z, dans le sens où ils ont presque le même comportement.

Ces registres sont stockés en RAM SYSTEM, comme suit :

7A00 registre Xa
7A08 registre Za
7A10 registre Ya
7A18 registre Ua
7A20 registre Va
7A28 registre Wa
7A30 registre Sa

Les registres Xa et Ya sont les plus importants, ceux qui contiendront les opérandes des différentes opérations, les autres registres sont utilisés par les calculs intermédiaires.

Un registre arithmétique peut représenter trois types d'informations, selon la valeur du 5ème octets, que l'on appellera caractéristique du registre.

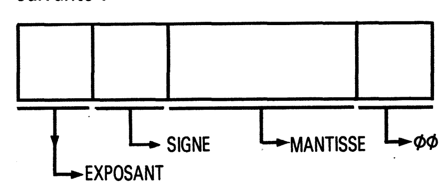
□	□	□	□	C	□	□	□
1	2	3	4	5	6	7	8

Du fait de l'équivalence entre les registres arithmétiques et statiques, nous utiliserons pour nos exemples ces derniers et en particulier la variable A qui commence en RAM en &7900. Pourquoi ? : car les registres arithmétiques sont toujours sollicités pour le stockage d'arguments, pour les calculs... ce qui les rend du point de vue du BASIC extrêmement instables. Par exemple lorsque vous frappez au clavier : STR$ (10+15) « ENTER », le registre arithmétique Xa va contenir successivement : le nombre 15 puis le nombre 25 et enfin la chaine « 25 ».

REPRESENTATION D'UN NOMBRE EN FLOTTANT :

Quand on a c<&B2, le registre contiendra un nombre flottant comprit entre −9.999999999 E99 et +9.999999999 E99

Ce nombre sera codé sur 8 octets de la manière suivante :



L'exposant est exprimé sur 8 bits signés donc entre −128 et +127. Quand on quitte l'intervalle (−99, +99) l'exposant est tronqué à l'affichage mais pas dans les calculs. Par exemple frappez A=1 puis POKE & 7900, 127. tapez alors A < ENTER > il s'affiche 1 E12, mais LOG A donne 127 : A contenait bien 1 E 127.

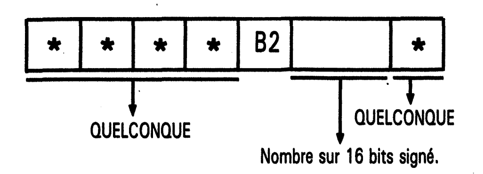
Le second octet donne le signe de la mantisse : si sa valeur est 00, le nombre sera positif, et négatif si sa valeur est 80.

La mantisse est codée en DCB sur 5 octets, capable donc de donner 10 chiffres significatifs. Le dernier octets appartient aussi à la mantisse et nous apparaît comme nul. En fait il est utilisé pour les calculs internes qui se font sur 12 chiffres significatifs. Ici aussi on peut obtenir des résultats curieux en ne respectant pas les règles du DCB. Par exemple tapez :

A=0, POKE & 7900, 1, 0, 255 puis A < ENTER > il s'affiche « ?? », qui représente le nombre 165, on peut le vérifier en frappant A*1.

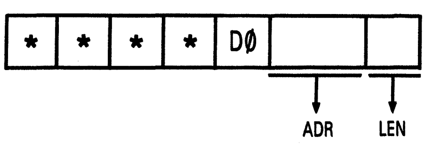
La mantisse est toujours comprise entre 0 et 1, par exemple le nombre 4573 sera représenter : 0.4573 E4.

REPRESENTATION D'UN NOMBRE ENTIER :
Il faut pour cela que c=B2. Les octets 5 et 6 servent à coder un nombre compris entre −32768 et +32767, sur 16 bits signés tous les autres octets sont utilisés.



REPRESENTATION D'UNE VARIABLE :
Dans le cas où c>&B2 (en général c=&DO) le registre va permettre d'adresser indirectement une chaine de caractères ou une autre variable.

Le registre est alors codé comme suit :



ADR : adresse de la chaine ou de la variable sur deux octets.

LEN : longueur de la chaine entre 0 et 80
LEN : vaut 88 si l'on adresse une autre variable, notez que ceci ne fonctionne pas avec les registres statiques.

Par exemple : faite POKE & 7904, & DO, & C3, & 4F, & OC à présent la variable A représente la chaine de 12 caractères qui débute en C34F. (Cette chaine est le message « NEW 0 CHECK »)

FONCTIONS SUR LES R.A :
Nous allons étudier dans ce chapitre la plupart des fonctions qui permettent de gérer les registres arithmétiques. Chaque fonction vous est donnée avec l'adresse de la routine suivie, si elle existe, du numéro de la macro-instruction qui correspond.
On notera BC, DE et HL les registres du microprocesseur X, Y et Z, afin de ne pas les confondrent avec les registres arithmétiques.

Notez qu'avant le premier appel de ces fonctions, il faut utiliser la macro 54 afin d'initialiser les registres B ou D avec la valeur 7A.

CHARGEMENT SUR 8 OCTETS :

fonction	adresse	macro	fonction	adresse	macro
LD S, Y	F701		LD S, X	F707	
LD Y, S	F715	68	LD X, S	F737	
LD Y, X	F70D	E6	LD X, Y	F73D	56
LD X, (BC)	F73F		LD Y, (BC)	F717	
LD S, (BC)	F703		LD (DE), X	F711	
LD (DE), (BC)	F741				

CHARGEMENT SUR 7 OCTETS (SANS L'EXPOSANT) :

fonction	adresse	macro	fonction	adresse	macro
LD7 X, Y	F71B		LD7 Y, X	F729	82
LD7 X, Z	F71F		LD7 Z, X	F72F	
LD7 X, (BC)	F721		LD7 Y, (BC)	F72B	
LD7 (DE), (BC)	F733				

ECHANGES :

fonction	adresse	macro	fonction	adresse	macro
EXC X, S	F7B5		EXC X, Y	F7B9	66
EXC X, (DE)	F7BB		EXC (BC), (DE)	F7BD	

MISE A ZERO SUR 8 OCTETS :

fonction	adresse	macro	fonction	adresse	macro
CL X	F757	EC	CL Y	F753	
CL (BC)	F75B				

MISE A ZERO SUR 7 OCTETS :

fonction	adresse	macro	fonction	adresse	macro
LD7 X, 0	F75F	76	LD7 Y, 0	F753	
LD7 Z, 0	F74B		LD7 W, 0	F74F	
LD7 (BC), 0	F761				

EXEMPLE :
Par exemple charger le registre arithmétique Xa avec la variable A .

```
CD7A     VMJ 7A     D=7A B=7A
4879     LDI B, 79  On charge BC avec l'adresse de
4A00     LDI C, 00  la variable statique A.
BEF721   SJP F721   LD X, (BC) charge le registre mathématique.
```

COMPARAISONS SUR LES R.A :

COMPARAISONS NUMERIQUES :
Pour comparer deux nombres, il faut tout d'abord les stocker dans les registres Xa et Ya.

Le type de la comparaison est stocké dans l'accumulateur de la manière suivante :

A=	Ø	Ø	Ø	Ø	=	>	<	<>
	b7	b8	b5	b4	b3	b2	b1	bØ
	128	64	32	16	8	4	2	1

Par exemple si l'on veut que l'opérateur de comparaison soit inférieur ou égal, on chargera A avec la valeur 2+8=10 (en décimal).
Puis il faut appeler la routine de comparaison par CALL & DOD2, au retour, si le test est vérifié on aura Xa=1 et le flag Z sera mis à 0, sinon on aura Xa=0 et Z=1.
Par exemple, nous allons écrire une routine qui met le plus grand des deux nombres contenu dans les variables statiques A et B, dans C :

```
CD 54    VMJ     54    B=7A, D=7A
4879     LDI B   79    BC pointe sur la variable statique A
```

4A00	LDI C	00	
BEF73F	SJP	F73F	Charge Xa avec le contenu de A
4A08	LDI C	08	BC pointe sur la variable statique B
BEF717	SJP	F717	Charge Ya avec le contenu de B
B506	LDI A	06	test Xa supérieur ou égal à Ya
BEDOD2	SJP	DOD2	appelle la routine de test
5879	LDI D	79	DE pointe sur la variable C
5A10	LDI E	10	
4879	LDI B	79	BC pointe sur la variable A
4A00	LDI C	00	
8902	BZR	+02	si le test est vrai, A> =B, on exécute le saut
4A08	LDI C	08	sinon BC pointe sur la variable B
BEF741	SJP	F741	(DE) = (BC), C = A si A> = B sinon C = B
9A	RTN		retour en BASIC.

COMPARAISON ALPHABETIQUE :

Les deux chaines sont stockées dans les registres Xa et Ya. Le type de la comparaison est stocké dans l'accumulateur comme pour une comparaison numérique. L'adresse de la routine est cette fois-ci DOF9. Au retour le registre Xa et le flag Z donnent les mêmes informations que lors des comparaisons numériques.

COMPARAISON A ZERO :

Quand le registre est formaté, si le troisième octet du registre a pour valeur 0, alors le registre est nul. Vous pouvez le vérifier en tappant A = 0, puis PEEK & 7902.
Par exemple comparer Xa à zéro est équivalent à : CP (&7A02), 0. On aura Z = 1 si Xa = 0 et Z = 0 autrement.

OPERATIONS SUR LES R.A :

Ici aussi pour chaque opération est donnée l'adresse d'entrée, le numéro de la MACRO-instruction si elle existe et les registres arithmétiques modifiés lors des calculs intermédiaires. Vous constaterez que le registre S est rarement modifié, et pourra donc contenir vos propres intermédiaires.

OPERATIONS DIADIQUES :

Xa=Xa+Ya	EFBA	FO	pas de registres modifiés
Xa=Xa−Ya	EFB6		pas de registres modifiés
Xa=Xa*Ya	F01A		pas de registres modifiés
Xa=Xa/Ya	F084	58	S et W non modifiés
Xa=Ya/Xa	F082		S et W non modifiés
Xa=Xa^Ya	F89C		tous les registres sont modifiés

OPERATIONS MONOADIQUES :

Xa=SQR(Xa)	FOE9		U, V, W et S non modifiés
Xa=Xa 2	FOE9		Y=X et S non modifié
Xa=1/Xa	FO80	6E	Y=X et W, S non modifiés
Xa=LNXa	F161		W=0 et S non modifié
Xa=LOGXa	F165		W=0 et S non modifié
Xa=EXPXa	F1CB		W=0 et S non modifié
Xa=10 Xa	F1D4		W=0 et S non modifié
Xa=SINXa	F3A2		tous les registres sont modifiés
Xa=COSXa	F391		tous les registres sont modifiés
Xa=TANXa	F39E		W=0 et S non modifié
Xa=ASNXa	F49A		W=0 et S non modifié
Xa=ACSXa	F492		W=0 et S non modifié
Xa=ATNXa	F496		Z=0 et S non modifié
Xa=DEGXa	F531		W non modifié
Xa=DMSXa	F564		tous les registres sont modifiés
Xa=ABSXa	F597		aucun registre modifié
Xa=SGNXa	F59D		aucun registre modifié
Xa=INTXa	F5BE		seul Y est modifié, Y=FRAC Xa
Xa=RNDXa	F5DD		tous les registres sont modifiés

Si un calcul conduit à un dépassement de capacité ou à une erreur, le flag C (carry) prend la valeur 1 et le registre du microprocesseur UH contient le numéro de l'erreur. En ce qui concerne les registres du microprocesseur, ils sont tous modifiés et l'on a B = 7A et D = 7A.

Lors de l'éxécution de ces opérations, tous les résultats se trouvent dans le registre arithmétique Xa et sont généralement normalisés. Dans le cas contraire il faut utiliser la macro D8 pour normaliser Xa. Mais je vous dois quelques explications sur le problème de normalisation :
Quand l'ordinateur effectue les calculs, la plupart du temps par l'intermédiaire de décalages, le résultat, n'est pas forcément conforme à la représentation interne des nombres, telle je vous l'ai décrite au premier chapitre. Par exemple au bon format interne : 0.123457 E21.

OPERATIONS SUR LES CHAINES DE CARACTERES :

On peut distinguer trois types d'opérations :
- Xa contient un nombre, le résultat est une chaine
- Xa contient une chaine, le résultat est un nombre
- on transforme une chaine en une autre chaine

1er CAS :

notez qu'il faut que (7894) contienne la valeur 10 hexa. Xa pointe la chaine résultat, qui se trouve dans le buffer de chaine à partir de l'adresse &7B10.
Xa$ = CHR$ Xa D9B1
Xa$ = STR$ Xa D9CF

2e CAS :

Xa pointe sur une chaine, le résultat est un nombre
Xa = VAL Xa$ D9D7
Xa = LEN Xa$ D9DDattention YL doit valoir 64 HEXA
Xa = ASC Xa$ D9DDattention YL doit valoir 60 HEXA

3e CAS :

C'est le plus compliqué, simulant la fonction LETF$, RIGHT$ et MID$

Il faut tout d'abord utiliser un speudo registre arithmétique qui pointera la chaine à traiter. Ce registre sera placé dans le bluffer BASIC à la suite de la pile des boucles FOR-NEXT. Pour connaitre l'adresse de la fin de cette pile, on utilise la valeur du pointeur (&7890) :

@=Adresse=&7A00+(&7890). De @ à @+7 on aura notre pseudo-registre arithmétique.

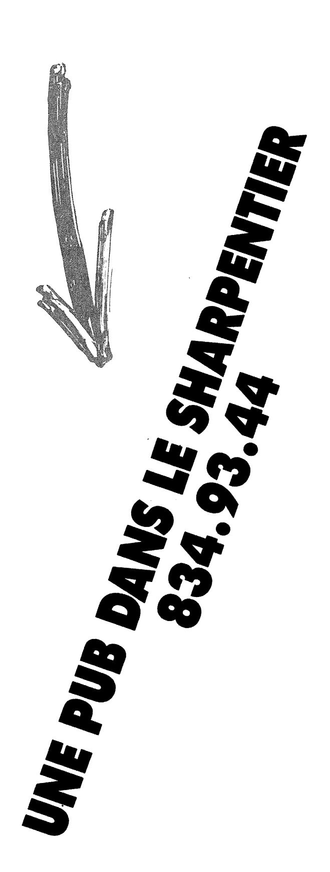
Le ou les paramètres numériques sont stockés dans les registres arithmétiques Xa, Ya. Après éxécution le registre Xa pointe la chaine résultat.

7A HEXA
Xa$=LEFT$($, Xa) D9F3 YL doit valoir 7A HEXA
Xa$=RIGHT$($, Xa) D9F3 YL doit valoir 02 HEXA
Xa$=MID$($, Xa, Ya) D9F3 YL doit valoir 7B HEXA

Voilà, j'espère qu'à partir de tous ces renseignements vous pourrez facilement exécuter des opérations à partir du langage machine.

Notez que pour les routines de traitement alphabétique les gains de temps sont moins spectaculaires que pour les routines de traitement alphabétique, surtout lors de l'appel des fonctions transcendantes.

Pascal ABRIVARD



RELOGEUR BIDI LIST

relogeur BIDI LIST DE M. FOJUD
(c) 1985

Tapez le programme BIDI du bulletin n° 13 à une adresse quelconque,
Entrez ce programme, faites RUN, tapez l'adresse à laquelle est implanté
Le programme original et ENTER, et après quelques secondes la transformation
est faite, vous pouvez utiliser ce programme par CALL (1re adresse).
Il ne faut utiliser ce relogeur qu'une fois sous peine de plantage.

```
100 INPUT "ADRESSE";A:D=A-&5CFO
110 B=&5F5D+D:GOSUB 300:POKE A+13,H,L:POKE A+278,H,L:POKE A+598,H,L
120 B=&5FD8+D:GOSUB 300:POKE A+167,H,L,:POKE A+368,H,L:POKE A+469,H,L:POKE
A+528,H,L : POKE A+673,H,L
130 B=&5FAB+D:GOSUB 300:POKE A+179,H,L:POKE A+339,H,L
140 B=&5F04+D:GOSUB 300:POKE A+210,H,L
150 B=&5FB7+D:GOSUB 300:POKE A+275,H,L:POKE A+291,H,L:POKE A+326,H,L:POKE
A+499,H,L
160 POKE A+587,H,L:POKE A+650,H,L
170 B=&5F97+D:GOSUB 300:POKE A+307,H,L:POKE A+617,H,L
180 B=&5F7D+D:GOSUB 300:POKE A+375,H,L:POKE A+509,H,L
190 B=&5D01+D:GOSUB 300:POKE a+610,H,L
200 B=&5D9B+D:GOSUB 300:POKE A+484,H,L
210 END
300 H=B/256:L=BAND &FF:RETURN
```

ERRATA PC 1500

Nous nous sommes aperçus d'une erreur flagrante mais qui nous avait échappée jusqu'à présent. Les # ont été oubliés dans la deuxième table des instructions ; n° 10, page 16. Il faut en mettre avant les parenthèses ouvrantes et seulement celles-ci.
Ex : case 01 table 2 SBC (X) devient SBC # (X)
mais case 4–1 table 2 SIN X reste SIN X
De plus, case 98, c'est bien sûr un LOP UL et non VL ce qui n'aurait aucun sens : VL, valant toujours 00.
On peut signaler aussi que les nouveaux codes prennent exactement le même nombre cycles que les anciens : exemple
table 1 case 00 SBC XL codé 00 : 6 cycles
table 2 case 00 SBC XL codé FD00 : 6 cycles

La première valeur est donnée par le "Technical reference manual", la seconde est calculée expérimentalement.

GESTION DU CLAVIER (2)

Comme promis, nous allons voir cette fois ce que fait exactement la routine E42C, puis ses erreurs et un programme créant un super Inkey, qui sortira toutes les touches appuyées, plus quelques utilitaires touchant au clavier.

DESASSEMBLAGE DE LA ROUTINE E4 2C.

La routine E42C se sert d'une table de conversion se trouvant en ROM. Elle se situe aux adresses &FE80 à &FEBF.
Cette table contient les codes ASCII des touches classées dans l'ordre PA croissant, IN décroissant.
Voir liste 3
rappel figure 1
Il est bien sûr indispensable pour saisir cet article de lire le premier concernant la gestion du clavier (Bulletin n° 11).

EXPLICATION INSTRUCTION PAR INSTRUCTION :

voir liste 4.
E42C : On met dans XI le poids faible du premier octet de la table de correspondance. XI sera incrémenté au fur et à mesure de la recherche.
E42E : A va servir pour la routine de scrutation, au début A = 1 donc on commence à tester PAO.
E430 : On le sauve dans Xh car A est modifié par la routine E41A, ainsi on sait toujours à quel niveau d'exploration on se trouve.
E431 : Lecture clavier par bloc de 8 touches suivant A.
E434 : Si Z = 1, c'est qu'aucune touche pour ce PA n'a été appuyée, on va donc tester le bloc PA suivant.
E436 : C = 0 pour l'addition, on fera donc A = XI + A + C = A = XI + A
E437 : A = 8, on incrémente XI de 8, XI pointant sur la table de
E439 : Conversion, car on a aucune touche appuyée pour ce PA, l'exploration continue avec le PA suivant, donc 8 touches plus loin, puisqu'un PA teste 8 touches : IN0 à IN7
E43A : On remet dans XI le résultat pour obtenir un suivi sur le tableau de correspondance.
E43B : On a sauvé dans Xh le numéro de l'exploration des PA, on le reprend dans A pour passer au suivant.
E43C : Fait un décalage vers la gauche d'un bit. A pointait le numéro de l'exploration, il pointera donc le suivant. Dans C, on retrouve le bit 7 de A.
E43D : Si le carry = 1, A valait &80, le 7e bit était allumé, les 8 bits ont déjà été testés, aucune touche n'est appuyée, on se branche en E441, qui se charge d'indiquer qu'aucune touche n'est appuyée.
E43F : Z = 1 si A = 0, donc si l'exploration est terminée.
Ce test est stupide, car si le programme vient là, c'est que C = 0 donc l'exploration n'est pas finie.
Il aurait été plus judicieux de mettre :
E43D : BCR E 430
E43F . NOP
E440 : NOP
ou même de supprimer les NOP d'où un gain de 2 octets.
E441 : Le programme vient ici si aucune touche n'est appuyée, A = 0, soit le code ASCII de NUL. Le carry est à 1 à cause du SLA. Un LDA VI gagne un octet.
E443 : Retour à l'appel.
E444 : Le programme vient ici s'il a détecté une touche appuyée.
On a dans XI le poids faible de l'adresse où en est l'exploration sur le tableau de correspondance ; dans A, on a la correspondance de IN. A est forcément différent de zéro. On va rechercher le premier bit allumé de A en partant de la gauche, d'où le SLA, C prend la valeur du bit 7 et A est décalé vers la gauche.
E445 : Si le carry est à un donc si on a trouvé la touche appuyée, on va en E44C prendre le code ASCII de la touche d'après le tableau de correspondance.
E447 : On incrémente la position du tableau de correspondance car on va tester le bit suivant.
E448 : Le programme teste si XI ⩽ %FF, on traite le bit suivant.
E44A : Si XI = &FF on met A à 0. Ce test est de nouveau stupide car si XI = &FF c'est qu'aucun bit n'est allumé, or s'il vient là, c'est qu'il y en a au moins 1. On aurait dû faire :
E448 : JR E444
E44A : NOP
E44B : NOP
D'où un gain de 2 octets (encore !).
E44C : On vient ici quand on a déterminé la touche appuyée. On a dans XI le poids faible de la position de la touche dans le tableau de correspondance. Le poids fort est mis dans Xh et vaut &FE.
E44E : Dans A le code ASCII pris du tableau
E44F : C = 0
E450 : Retour de l'appel.
Soit donc en résumé, au retour :
Si C = 1, A = 0, pas de touches appuyées,
UL = &FF
X = &80C0
A = &00
UH et Y sont inchangés.
Si C = 0, A <> 0, une touche dont le code ASCII est dans A est appuyée
UL = &FF
X = adresse du code de A
dans la table de correspondances
Xh = &FE
A = valeur en ASCII

UH et Y sont inchangés.
Voici pour les amateurs, la même routine sans parties stupides. Elle fait 30 octets, alors que l'original en occupe 37, soit un gain de 11,1 %.
Les 16 Ko de ROM seraient réduits en 14,2 Ko si le gain était proportionnel à la place mémoire.
La nouvelle routine est relogeable. Liste 5.
Pour finir, voici une astuce : si vous voulez dans A, non le code ASCII, mais le code ASCII "SHIFTE", sachez que la table de correspondance commence en &FECO et finit en &FEFF.
Pour l'utiliser, faire donc :
LDI XI, &CO indique que le tableau commence en &FECO SPJ &E42E saute le LDI XI, &80 du programme initial.
Nous verrons la fois prochaine, le SUPER INKEY$ et les erreurs "physiques" du clavier.

Marc GIRONDOT

LISTE 3

```
ADSS: PA   IN   PK  C
FE80: PA0  IN7  0B
FE81: PA0  IN6  4E  N
```



```
FE82:  PA0  IN5  59  Y
FE83:  PA0  IN4  01
FE84:  PA0  IN3  48  H
FE85:  PA0  IN2  38  8
FE86:  PA0  IN1  35  5
FE87:  PA0  IN0  32  2
FE88:  PA1  IN7  09
FE89:  PA1  IN6  58  X
FE8A:  PA1  IN5  57  W
FE8B:  PA1  IN4  11
FE8C:  PA1  IN3  53  S
FE8D:  PA1  IN2  0F
FE8E:  PA1  IN1  2D  -
FE8F:  PA1  IN0  2E  .
FE90:  PA2  IN7  30  0
FE91:  PA2  IN6  4D  M
FE92:  PA2  IN5  55  U
FE93:  PA2  IN4  15
FE94:  PA2  IN3  4A  J
FE95:  PA2  IN2  37  7
FE96:  PA2  IN1  34  4
FE97:  PA2  IN0  31  1
FE98:  PA3  IN7  0D
FE99:  PA3  IN6  28  (
FE9A:  PA3  IN5  49  I
FE9B:  PA3  IN4  16
FE9C:  PA3  IN3  4B  K
FE9D:  PA3  IN2  4F  O
FE9E:  PA3  IN1  4C  L
FE9F:  PA3  IN0  29  )
FEA0:  PA4  IN7  19
FEA1:  PA4  IN6  43  C
FEA2:  PA4  IN5  45  E
FEA3:  PA4  IN4  12
FEA4:  PA4  IN3  44  D
FEA5:  PA4  IN2  2F  /
FEA6:  PA4  IN1  2A  *
FEA7:  PA4  IN0  2B  +
FEA8:  PA5  IN7  20
FEA9:  PA5  IN6  56  V
FEAA:  PA5  IN5  52  R
FEAB:  PA5  IN4  13
FEAC:  PA5  IN3  46  F
FEAD:  PA5  IN2  50  P
FEAE:  PA5  IN1  08
FEAF:  PA5  IN0  3D  =
FEB0:  PA6  IN7  02
FEB1:  PA6  IN6  5A  Z
FEB2:  PA6  IN5  51  Q
FEB3:  PA6  IN4  1B
FEB4:  PA6  IN3  41  A
FEB5:  PA6  IN2  18
FEB6:  PA6  IN1  1F
FEB7:  PA6  IN0  0C
FEB8:  PA7  IN7  0A
FEB9:  PA7  IN6  42  B
FEBA:  PA7  IN5  54  T
FEBB:  PA7  IN4  14
FEBC:  PA7  IN3  47  G
FEBD:  PA7  IN2  39  9
FEBE:  PA7  IN1  36  6
FEBF:  PA7  IN0  33  3
```

LISTE 4

```
E42C:  LDI  XL,&80
E42E:  LDI  A,&01
E430:  STA  XH
E431:  SJP  &E41A
E434:  BZR  +&0E  "E444"
E436:  REC
E437:  LDI  A,&08
E439:  ADC  XL
E43A:  STA  XL
E43B:  LDA  XH
E43C:  SHL
E43D:  BCS  +&02   "E441"
E43F:  BZR  -&11   "E430"
E441:  LDI  A,&00
E443:  RTN
E444:  SHL
E445:  BCS  +&05   "E44C"
E447:  INC  XL
E448:  BCR  -&06   "E444"
E44A:  BCH  -&0B   "E441"
E44C:  LDI  XH,&FE
E44E:  LDA  (X)
E44F:  REC
E450:  RTN
```

LISTE 5

```
40C5:  LDI  XL,&7F
40C7:  LDI  A,&01
40C9:  STA  XH
40CA:  SJP  &E41A
40CD:  BZR  +&0B  "40DA"
40CF:  REC
40D0:  LDI  A,&08
40D2:  ADC  XL
40D3:  STA  XL
40D4:  LDA  XH
40D5:  SHL
40D6:  BCR  -&0F   "40C9"
40D8:  LDA  UL
40D9:  RTN
40DA:  INC  XL
40DB:  SHL
40DC:  BCR  -&04   "40DA"
40DE:  LDI  XH,&FE
40E0:  LDA  (X)
40E1:  REC
40E2:  RTN
```

LISTE 6

```
ADSS:  PA   IN   PK  C
FEC0:  PA0  IN7  5B  [
FEC1:  PA0  IN6  6E  n
FEC2:  PA0  IN5  79  y
FEC3:  PA0  IN4  01
FEC4:  PA0  IN3  68  h
FEC5:  PA0  IN2  38  8
FEC6:  PA0  IN1  35  5
FEC7:  PA0  IN0  32  2
FEC8:  PA1  IN7  09
FEC9:  PA1  IN6  78  x
FECA:  PA1  IN5  77  w
FECB:  PA1  IN4  21  !
FECC:  PA1  IN3  73  s
FECD:  PA1  IN2  0F
FECE:  PA1  IN1  2C  ,
FECF:  PA1  IN0  2E  .
FED0:  PA2  IN7  30  0
FED1:  PA2  IN6  6D  m
FED2:  PA2  IN5  75  u
FED3:  PA2  IN4  25  %
FED4:  PA2  IN3  6A  j
FED5:  PA2  IN2  37  7
FED6:  PA2  IN1  34  4
FED7:  PA2  IN0  31  1
FED8:  PA3  IN7  0D
FED9:  PA3  IN6  3C  <
FEDA:  PA3  IN5  69  i
FEDB:  PA3  IN4  26  &
FEDC:  PA3  IN3  6B  k
FEDD:  PA3  IN2  6F  o
FEDE:  PA3  IN1  6C  l
FEDF:  PA3  IN0  3E  >
FEE0:  PA4  IN7  19
FEE1:  PA4  IN6  63  c
FEE2:  PA4  IN5  65  e
FEE3:  PA4  IN4  22  "
FEE4:  PA4  IN3  64  d
FEE5:  PA4  IN2  3F  ?
FEE6:  PA4  IN1  3A  :
FEE7:  PA4  IN0  3B  ;
FEE8:  PA5  IN7  5E  ^
FEE9:  PA5  IN6  76  v
FEEA:  PA5  IN5  72  r
FEEB:  PA5  IN4  23  #
FEEC:  PA5  IN3  66  f
FEED:  PA5  IN2  70  p
FEEE:  PA5  IN1  1D
FEEF:  PA5  IN0  40  @
FEF0:  PA6  IN7  02
FEF1:  PA6  IN6  7A  z
FEF2:  PA6  IN5  71  q
FEF3:  PA6  IN4  1B
FEF4:  PA6  IN3  61  a
FEF5:  PA6  IN2  1A
FEF6:  PA6  IN1  1E
FEF7:  PA6  IN0  1C
FEF8:  PA7  IN7  5D  ]
FEF9:  PA7  IN6  62  b
FEFA:  PA7  IN5  74  t
FEFB:  PA7  IN4  24  $
FEFC:  PA7  IN3  67  g
FEFD:  PA7  IN2  39  9
FEFE:  PA7  IN1  36  6
FEFF:  PA7  IN0  33  3
```

PC 1251

DEVELOPPEMENTS LIMITES

Les dl, tout le monde connaît ou presque et sait combien cela est ennuyeux à calculer. Ce programme est donc capable de calculer F (G(x)) où F et G sont des dl.
Mode d'emploi :
Tapez RUN

Y et U sont les dl. S'ils figurent dans les DATAs, tapez leurs noms. Autrement ENTER, puis les entrer à la main (les coefficients). Ensuite, le PC calculera et affichera pour vous les coefficients deY(U).

Y. RAVELOMANATSOA.

```
1:"*DL G(F(X))*
2:"COS" DATA 1,0,-.5,0
   ,1/24,0,-1/720
3:"SIN" DATA 0,1,0,-1/
   6,0,1/120,0
4:"EXP" DATA 1,1,.5,1/
   6,1/24,1/120,1/720
5:"√(1+X)" DATA 1,.5,-
   1/8,1/16,-5/128,7/25
   6,-63/2816
6:"√(1-X)" DATA 1,-.5,
   -1/8,-1/16,-5/128,-7
   /256,-63/2816
7:"1/(1+X)" DATA 1,-1,
   1,-1,1,-1,1
8:"1/(1-X)" DATA 1,1,1
   ,1,1,1,1
9:"LN(1+X)" DATA 0,1,-
   1/2,1/3,-1/4,1/5,-1/
   6
10:"LN(1-X)" DATA 0,-1,
   -1/2,-1/3,-1/4,-1/5,
   -1/6
11:"SH" DATA 0,1,0,1/6,
   0,1/120,0
12:"CH" DATA 1,0,1/2,0,
   1/24,0,1/720
13:"ACS" DATA .5π,-1,0,
   -1/6,0,-3/40,0
14:"ASN" DATA 0,1,0,1/6
   ,0,3/40,0
15:"TAN" DATA 0,1,0,1/3
   ,0,2/15,0,0
40:INPUT "ORDR(<=6) ";N
   : DIM U(N),R(N),Y(N)
   ,S(N,N)
41:INPUT "DATA Y?";W$:
   RESTORE W$: FOR I=0
   TO N: READ Y(I):
   NEXT I: GOTO 43
42:FOR I=0 TO N: INPUT
   "Y";Y(I): NEXT I
43:INPUT "DATA U?";W$:
   RESTORE W$: FOR I=0
   TO N: READ U(I):
   NEXT I: GOTO 46
44:FOR I=0 TO N: INPUT
   "U";U(I): NEXT I
46:R(0)=Y(0): FOR I=0
   TO N:R(I)=R(I)+U(I)*
   Y(1): NEXT I
48:FOR J=0 TO N:S(0,J)=
   U(0)*U(J): NEXT J
49:FOR I=1 TO N: FOR J=
   I TO N:S(I,J)=U(I)*U
   (J-I): NEXT J: NEXT
   I
50:FOR I=0 TO N: FOR J=
   0 TO I:R(I)=R(I)+S(J
   ,I)*Y(2): NEXT J:
   NEXT I

53:H=3: IF N<=2 THEN 80
54:"F" FOR I=0 TO N:
   FOR J=0 TO N-1:S(N,I
   )=S(N,I)+S(J,I):
   NEXT J: NEXT I
56:FOR J=0 TO N:S(0,J)=
   S(N,J)*U(0): NEXT J
58:FOR I=1 TO N: FOR J=
   I TO N:S(I,J)=U(I)*S
   (N,J-I): NEXT J:
   NEXT I
70:FOR I=1 TO N: FOR J=
   0 TO I-1:S(I,J)=0:
   NEXT J: NEXT I
74:FOR I=0 TO N: FOR J=
   0 TO I:R(I)=R(I)+S(J
   ,I)*Y(H): NEXT J:
   NEXT I
76:H=H+1: IF H<=N THEN
   "F"
80:FOR K=0 TO N:G=R(K):
   GOSUB 900: PRINT "CO
   EF";K;"=";B;"/";E:
   NEXT K: END
99:"***** END *****
900:"Z" AREAD G:H=G:I=
   INT H:A=1:B=I:D=0:E=
   1: IF H-I=0 PRINT G:
   RETURN
903:H=1/(H-I):I= INT H:C
   =B*I+A:A=B:B=C:F=E*I
   +D:D=E:E=F: IF ABS (
   1-C/FG)-E-9 THEN 903
904:PRINT B;"/";E:
   RETURN
```



UTILITAIRES GRAPHIQUES

Ces quatres petits programmes en LM destinés aux PC 1251/55/45 agrémenteront sans aucun doute vos utilitaires et jeux préférés par un affichage original.

Les trois premiers sont relogeables alors que le dernier demande des modifications faciles et pour être transplantés (adresse du tampon de l'afficheur). On peut les rentrer à l'aide, par exemple, du moniteur paru dans le numéro 8 ou bien celui paru dans le numéro 12. Le « pgm 5 » est un programme d'essai permettant de juger du résultat. Vérifiez bien vos codes machines avant de l'exécuter. Dans le cas du PC 1245 n'oubliez pas que son afficheur étant plus court que celui des PC 1251/55, il faut adapter en conséquence ces routines. Voici maintenant une brève descriptions de celles-ci :

Programme 1

permet de faire défiler l'afficheur de droite à gauche.
Faire WAIT nn puis PRINT « TEXTE » et CALL & C000 (ou adresse de relocation) dans un programme Basic.
En &C03C et &C03F se trouvent les paramètres de la boucle d'attente.

Programme 2

effectue un « scrolling » vertical ; de bas en haut.
Faire WAIT nn : PRINT « TEXTE » et CALL & C045.
&C06D et &C070 sont les paramètres de la boucle de temporisation.

Programme 3

défilement central : compression vers le centre.
WAIT nn : PRINT « TEXTE » : CALL &C076
&C09C et &C09F : paramètres d'attente.

Programme 4

ceci est inédit et permet d'afficher un texte d'une... Mais regardez plutôt.
WAIT 0 : PRINT « TEXTE » : CALL &C130
On modifie la durée de la boucle d'attente par un changement des valeurs en &C189 et &C18C.

Note : On utilise la Ram qui va de &C0B0 à &C12B comme buffer d'afficheur ; il faut en tenir compte pour reloger cette routine vers laquelle va ma préférence.
En plus des Dumps (pgm 6 et pgm7), vous pouvez étudier les désassemblages qui sont commentés. Voilà vous savez presque tout, il ne vous reste plus qu'à épater les autres cristaux liquides moins chanceux.

C. WOLFHUGEL.

pgm 1

```
C000 02  LIA   77
C002 34  PUSH
C003 F1  CAL 11E0
C005 10  LIDP  F800      B = caractère à l'extrême gauche
C008 57  LDD
C009 DA  EXAB
C00A 02  LIA   F8
C00C 85  LP05
C00D DB  EXAM
C00E 02  LIA   00        X = &F800
C010 84  LP04
C011 DB  EXAM
C012 02  LIA   F7
C014 87  LP07
C015 DB  EXAM            Y = &F7FF
C016 02  LIA   FF
C018 86  LP06
C019 DB  EXAM
C01A 02  LIA   3A
C01C 34  PUSH
C01D 24  IXL             On décale les 12 premiers caractères
C01E 26  IYS
C01F 2F  LOOP  03 (C01D)
C021 11  LIDL  7B
C023 57  LDD
C024 11  LIDL  3B        Décalage d'un bord
C026 52  STD
C027 02  LIA   7B
C029 84  LP04            X = &F87B
C02A DB  EXAM
C02B 02  LIA   7C
C02D 86  LP06            Y = &F87C
C02E DB  EXAM
C02F 02  LIA   3A
C031 34  PUSH
C032 25  DXL             Décalage les 12 derniers caractères
C033 27  DYS
C034 2F  LOOP  03 (C032)
C036 10  LIDP  F840
C039 DA  EXAB            B va à droite
C03A 52  STD
C03B 02  LIA   20
C03D 34  PUSH
C03E 4E  WAIT  FF        Boucle d'attente
C040 2F  LOOP  03 (C03E)
C042 2F  LOOP  3E (C005)
C044 37  RTN
```



pgm 2

```
C045 F1  CAL 11E0
C047 02  LIA  08
C049 34  PUSH
C04A 02  LIA  3B
C04C 34  PUSH
C04D 02  LIA  F7
C04F 85  LP05
C050 DB  EXAM
C051 02  LIA  FF             X=Y=&F7FF
C053 84  LP04
C054 DB  EXAM
C055 F1  CAL 1177
C057 24  IXL
C058 D1  RC
C059 D2  SR
C05A 26  IYS                 Décalage 12 caractères
C05B 2F  LOOP  05 (C057)
C05D 04  IX
C05E 04  IX
C05F 04  IX
C060 04  IX
C061 F1  CAL 1177
C063 02  LIA  3B
C065 34  PUSH
C066 24  IXL
C067 D1  RC
C068 D2  SR                  12 derniers caracteres
C069 26  IYS
C06A 2F  LOOP  05 (C066)
C06C 02  LIA  A0             1er paramètre d'attente
C06E 34  PUSH
C06F 4E  WAIT  FF            2e paramètre d'attente
C071 2F  LOOP  03 (C06F)
C073 2F  LOOP  2A (C04A)
C075 37  RTN
```

pgm 3

```
C076 F1  CAL 11E0
C078 02  LIA  3C
C07A 34  PUSH
C07B 02  LIA  F8
C07D 85  LP05                X=&F83B
C07E DB  EXAM
C07F 02  LIA  3B
C081 84  LP04                Y=&F83C
C082 DB  EXAM
C083 F1  CAL 1177
C085 06  IY
C086 02  LIA  3A             Décalage
C088 34  PUSH
C089 25  DXL
C08A 27  DYS
C08B 2F  LOOP  03 (C089)
C08D 02  LIA  7C             Y=&F87C
C08F 86  LP06
C090 DB  EXAM
C091 F1  CAL 1172            x=&F87B
C093 05  DX
C094 02  LIA  3A
C096 34  PUSH
C097 25  DXL                 Décalage
C098 27  DYS
C099 2F  LOOP  03 (C097)
C09B 02  LIA  30
C09D 34  PUSH
C09E 4E  WAIT  FF            Boucle d'attente
C0A0 2F  LOOP  03 (C09E)
C0A2 10  LIDP  F800
C0A5 02  LIA  00
C0A7 52  STD
C0A8 11  LIDL  40
C0AA 52  STD
C0AB 2F  LOOP  31 (C07B)
C0AD 37  RTN
```

pgm 4

```
C130 F1  CAL 11E0
C132 02  LIA  F7
C134 85  LP05
C135 DB  EXAM
C136 02  LIA  FF             X=&F7FF
C138 84  LP04
C139 DB  EXAM
C13A 02  LIA  C0
C13C 87  LP07
C13D DB  EXAM
C13E 02  LIA  AF             Y=&C0AF
C140 86  LP06
C141 DB  EXAM
C142 02  LIA  7B
C144 34  PUSH
C145 24  IXL                 Sauve afficheur de &C0B0 à &C12B
C146 26  IYS
C147 2F  LOOP  03 (C145)
C149 F1  CAL 1177
C14B 06  IY
C14C 02  LIA  3B
C14E 34  PUSH
C14F 02  LIA  00
C151 27  DYS
C152 2F  LOOP  02 (C151)     Vide l'affichage
C154 02  LIA  3B
C156 34  PUSH
C157 07  DY
C158 07  DY
C159 07  DY
C15A 07  DY
C15B 02  LIA  00
C15D 27  DYS
C15E 2F  LOOP  02 (C15D)
C160 02  LIA  90             Boucle
C162 34  PUSH
C163 02  LIA  C0
C165 85  LP05
C166 DB  EXAM                X=&C0AF
C167 02  LIA  AF
C169 84  LP04
C16A DB  EXAM
C16B 02  LIA  F7
C16D 87  LP07
C16E DB  EXAM                Y=&F7FF
C16F 02  LIA  FF
C171 86  LP06
C172 DB  EXAM
C173 02  LIA  7B
C175 34  PUSH
C176 A0  LP20
C177 24  IXL
C178 DB  EXAM                Le caractère en Ram est-il égal
C179 F1  CAL 1195            à celui de l'afficheur ?
```




```
C17B F1   CAL 1172            Oui : ne rien faire
C17D 24   IXL                 Non : l'incrémenter
C17E A0   LP20
C17F C7   CPMA
C180 38   JRZP   02 (C183)
C182 42   INCA
C183 26   IYS
C184 F1   CAL 1195
C186 2F   LOOP   11 (C176)
C188 02   LIA    10
C18A 34   PUSH
C18B 4E   WAIT   FF           Boucle d'attente
C18D 2F   LOOP   03 (C18B)
C18F 2F   LOOP   2D (C163)
C191 37   RTN
```

pgm 5

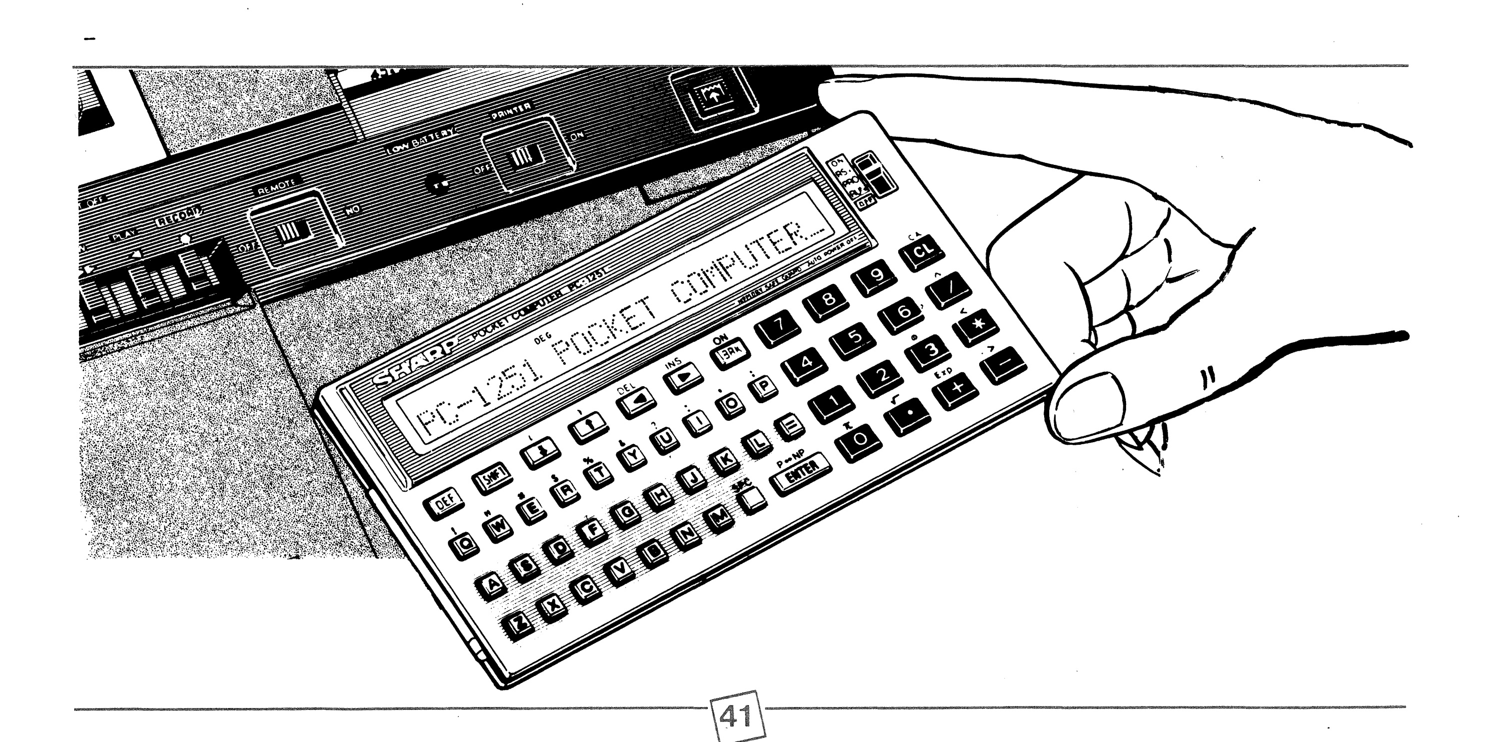
```
1:"PGM D ESSAI:DEF A
5:PRINT "UTILITAIRES G
  RAPHIQUES!!": RETURN
10:"A" WAIT 128: GOSUB
  5: CALL &C000
15:GOSUB 5: CALL &C045
20:GOSUB 5: CALL &C076
25:WAIT 0: GOSUB 5:
  CALL &C130
30:END
```

pgm 6

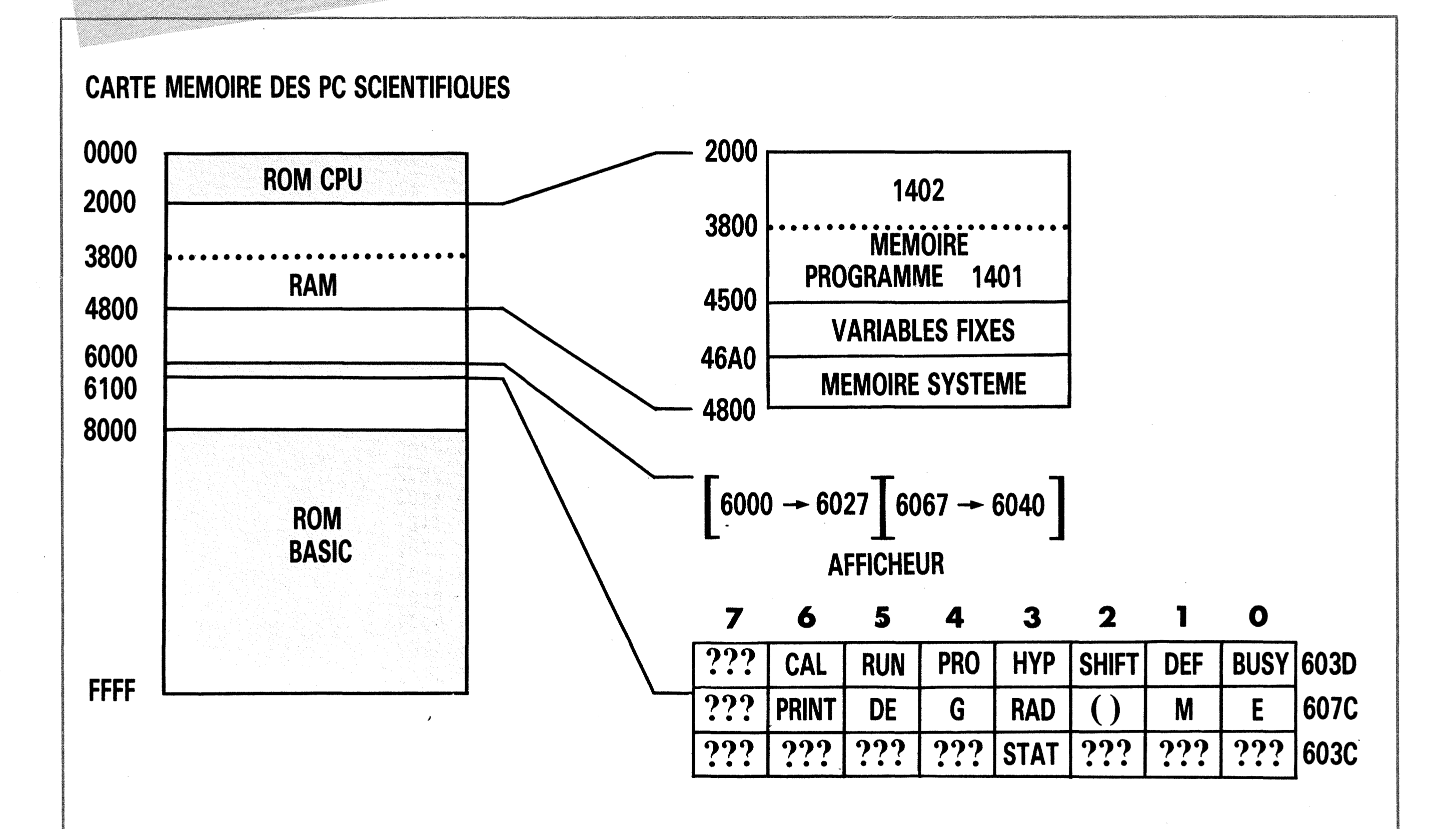
```
C000 0277 34F1 E010 F800
C008 57DA 02F8 85DB 0200
C010 84DB 02F7 87DB 02FF
C018 86DB 023A 3424 262F
C020 0311 7B57 113B 5202
C028 7B84 DB02 7C86 DB02
C030 3A34 2527 2F03 10F8
C038 40DA 5202 2034 4EFF
C040 2F03 2F3E 37F1 E002
C048 0834 023B 3402 F785
C050 DB02 FF84 DBF1 7724
C058 D1D2 262F 0504 0404
C060 04F1 7702 3B34 24D1
C068 D226 2F05 02A0 344E
C070 FF2F 032F 2A37 F1E0
C078 023C 3402 F885 DB02
C080 3B84 DBF1 7706 023A
C088 3425 272F 0302 7C86
C090 DBF1 7205 023A 3425
C098 272F 0302 3034 4EFF
C0A0 2F03 10F8 0002 0052
C0A8 1140 522F 3137 0000
```

pgm 7

```
C130 F1E0 02F7 85DB 02FF
C138 84DB 02C0 87DB 02AF
C140 86DB 027B 3424 262F
C148 03F1 7706 023B 3402
C150 0027 2F02 023B 3407
C158 0707 0702 0027 2F02
C160 0290 3402 C085 DB02
C168 AF84 DB02 F787 DB02
C170 FF86 DB02 7B34 A024
C178 DBF1 95F1 7224 A0C7
C180 3802 4226 F195 2F11
C188 0210 344E FF2F 032F
C190 2D37 0000 0000 0000
```



MEMORY MAP PC 140X



7	6	5	4	3	2	1	0	
???	CAL	RUN	PRO	HYP	SHIFT	DEF	BUSY	603D
???	PRINT	DE	G	RAD	( )	M	E	607C
???	???	???	???	STAT	???	???	???	603C

de

0000 à 1FFF ROM CPU

2000 à 37FF Rien ou RAM utilisateur pour PC1402

3800 à 45CF RAM utilisateur

45D0 à 469F Variables fixes

46A0 à 47FF Mémoire système 1

4800 à 5FFF Rien

6000 à 6FFF Afficheur

6100 à 7FFF Rien

8000 à FFFF ROM BASIC





Ceci est un petit utilitaire en LM qui transformera votre PC 1401/02 en un chronomètre au 100e de seconde.

Après avoir tapé le programme Basic, exécutez-le par RUN ENTER. Puis sauvegardez la partie LM par :
CSAVEM "CHRONO";&4000,&4136 ENTER
Vous pouvez d'ailleurs effacer le Basic par NEW ENTER. Ensuite le chrono se lance par CALL &4000 ENTER.
Les touches d'utilisation sont les suivantes :

- ▶ départ
- ◀ stop
- ↓ temps intermédiaires
- C-CE remise à zéro
- BASIC retour au Basic

Pour régler la précision du chronomètre, il faut modifier (par POKE) les valeurs du paramètre des WAITS aux adresses suivantes :
&4004 pour le général
&4051 pour les 100e
&4062 pour les 10e
&4073 pour les secondes
&4084 pour les dizaines de sec
&4095 pour les minutes
&40A6 pour les dizaines de mn

Si l'on veut récupérer le dernier résultat après une extinction de la machine, faire :
CALL &4015 ENTER

Etudiez bien ce programme, il vous donnera une foule de renseignements.
A vos marques...

C. MUNCH

```
10:REM CHRONO 100 EME
20:REM (C) C MUNCH
30:REM & LE CLUB
100:Y=&4100,X=&80CF:FOR
    I=1 TO 11:POKE Y,
    PEEK (X+4),PEEK (X+3
    ),PEEK (X+2),PEEK (X
    +1),PEEK X:X=X+5,Y=Y
    +5:NEXT I
4000:POKE &4000,&10,&40
    ,&B5,&02,&00,&52,&
    11,&B8,&52,&11,&BE
    ,&52,&11,&C1,&52,&
    11
4010:POKE &4010,&C7,&52
    ,&11,&CA,&52,&E5,&
    A2,&78,&40,&B0,&E4
    ,&8F,&67,&4B,&39,&
    1F
4020:POKE &4020,&67,&27
    ,&38,&B2,&67,&15,&
    29,&0D,&10,&40,&D0
    ,&02,&60,&52,&E4,&
    8F
4030:POKE &4030,&67,&03
    ,&28,&09,&10,&40,&
    D0,&02,&00,&52,&2C
    ,&03,&4E,&0C,&67,&
    10
4040:POKE &4040,&39,&27
    ,&78,&40,&B0,&10,&
    40,&CA,&57,&74,&05
    ,&52,&67,&32,&38,&
    05
4050:POKE &4050,&4E,&FA
    ,&2D,&2B,&02,&00,&
    52,&11,&C7,&57,&74
    ,&05,&52,&67,&32,&
    38

4060:POKE &4060,&05,&4E
    ,&B1,&2D,&3C,&02,&
    00,&52,&11,&C1,&57
    ,&74,&05,&52,&67,&
    32
4070:POKE &4070,&38,&05
    ,&4E,&05,&2D,&4D,&
    02,&00,&52,&11,&BE
    ,&57,&74,&05,&52,&
    67
4080:POKE &4080,&1E,&38
    ,&05,&4E,&05,&2D,&
    5E,&02,&00,&52,&11
    ,&B8,&57,&74,&05,&
    52
4090:POKE &4090,&67,&32
    ,&38,&05,&4E,&00,&
    2D,&6F,&02,&00,&52
    ,&11,&B5,&57,&74,&
    05
4100:POKE &40A0,&52,&67
    ,&1E,&38,&05,&4E,&
    00,&2D,&80,&37,&00
    ,&00,&00,&00,&00,&
    00
4110:POKE &40B0,&00,&04
    ,&85,&10,&41,&00,&
    18,&11,&00,&18,&11
    ,&32,&18,&11,&00,&
    18
4120:POKE &40C0,&11,&00
    ,&18,&11,&32,&18,&
    11,&00,&18,&11,&00
    ,&18,&00,&27,&85,&
    10
4130:POKE &40D0,&60,&00
    ,&19,&4E,&75,&37,&
    00,&00
```

ASTUCES ASTUCES

Bien que la tendance actuelle soit au LM, les astuces que je vous propose font essentiellement appel au Basic ; il est en effet fort peu probable que les membres du Club soient tous des inconditionnels de la « nuit-blanche-pour-optimiser-la-routine-LM-destinée-à-... » L'ensemble est peut-être un peu long mais je n'ai pas pu éviter un certain côté didactique : déformation professionnelle (je suis enseignant).



COMMANDE MERGE SUR PC-1401

Rappelons que cette commande permet de charger un programme (PROG2) à partir d'une cassette, sans effacer le programme (PROG1) déjà résident en mémoire (ce qui est inévitable à l'exécution de la commande "CLOAD"). Il suffit pour cela de masquer PROG1 en faisant pointer PDP à la fin de celui-ci (1) ; on charge alors PROG2 (2) puis on réinitialise PDP (3) :

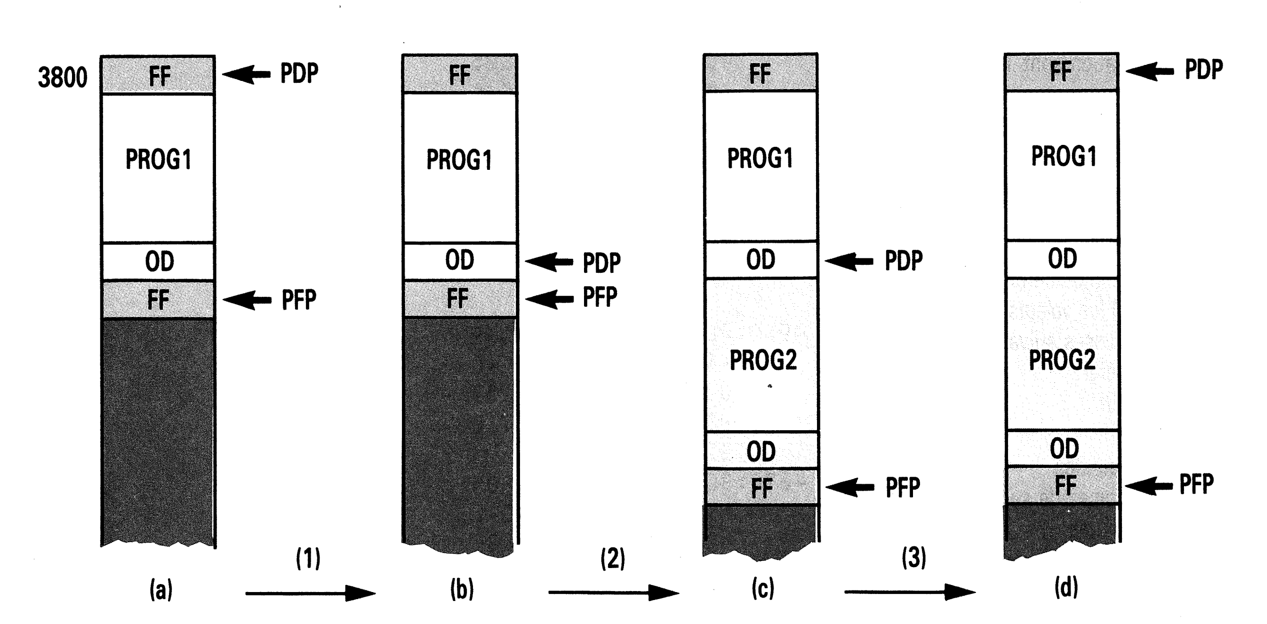


Figure 2

```
(1) A = PEEK &46E3 + 256 * PEEK &46E4
- 1
POKE &46E1 , A - 256 * INT (A/256) , INT
(A/256)
(2) CLOAD "PROG2"
(3) POKE &46E1 , &00 , &38
```

Remarques :

- Il faut veiller à ce que tous les numéros de lignes de PROG2 soient supérieurs à ceux de PROG1.
- Si vous éteignez la machine sans avoir réinitialisé PDP ((b) et (c)), vous provoquez un "ALL RESET".
- Si vous effectuez la commande "NEW" dans la même situation, le système va mettre à 255 l'octet pointé par PDP. N'oubliez pas de le remettre à 13 avant la réinitialisation (POKE A, &OD).



INSTRUCTION ERASE

Cette instruction permet de gagner de la place en mémoire en libérant le dernier tableau déclaré s'il vous est devenu inutile.
10000 "ERASE" P = &46FC : A = PEEK P + 256 * PEEK (P+1) : IF A = &45DO THEN RETURN
10010 A = 256 * PEEK (A+2) + PEEK (A+3) + 4 : POKE P, A − 256 * INT (A/256), INT (A/256) : RETURN

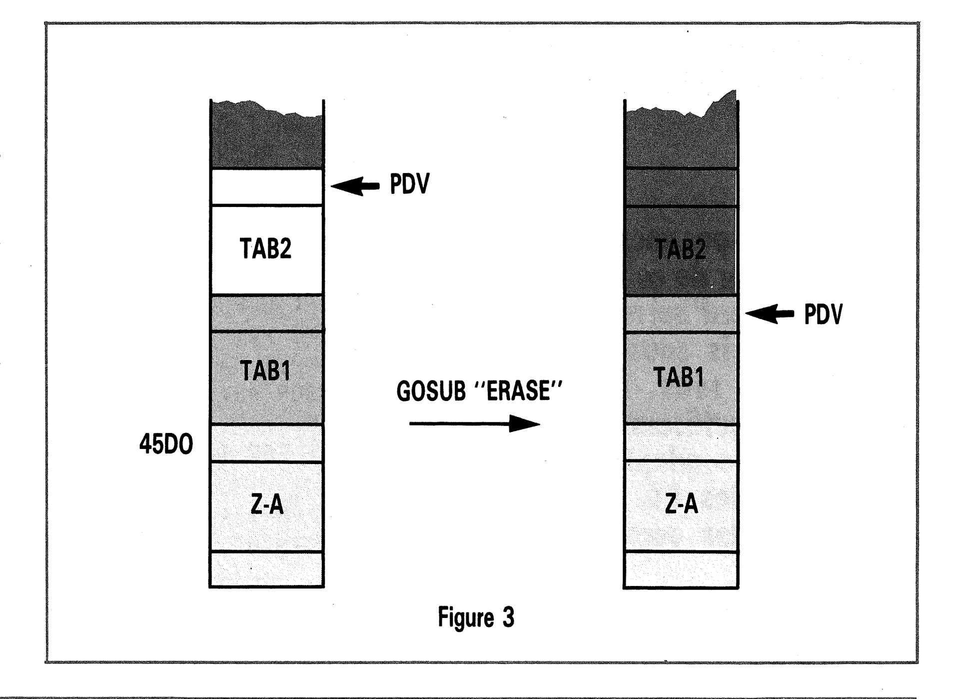


Figure 3



EXTENSION MEMOIRE

S'il vous manque quelques octets, et si toutes les variables fixes (Z à A) ne vous sont pas indispensables, vous pouvez gagnez jusqu'à 208 octets :
POKE &46FC, &AO, &46
Faites alors "MEM" : 3742 !! Mais vous n'avez plus une seule variable fixe. Sans aller jusqu'à cette solution extrême, vous pouvez gagner 8 octets par variable (en commençant par Z) :

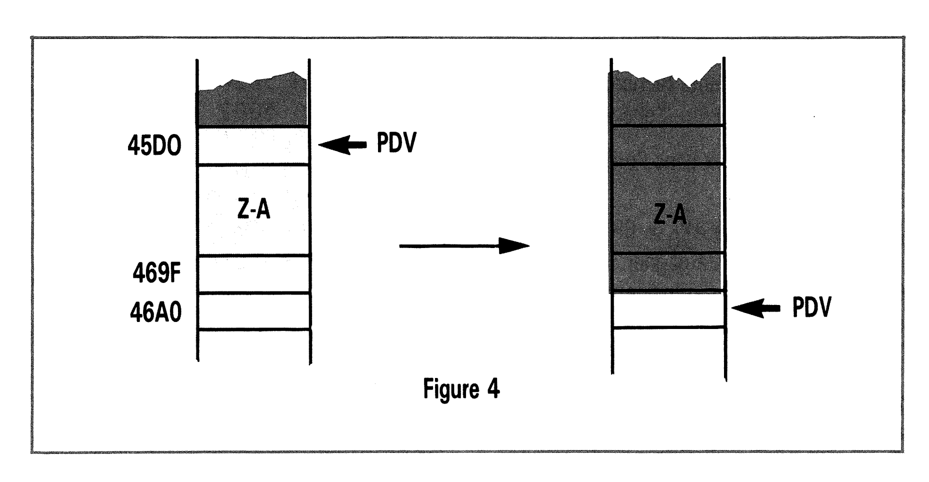


Figure 4

ANNEXE

Un peu de LM tout de même...
De tous ces trucs, le plus intéressant semble bien être la simulation de la commande "MERGE". Cette commande, pourtant bien utile, est bizarrement absente sur le PC-1401 alors qu'elle est présente sur tous les autres pockets de SHARP. Si l'on y fait souvent appel, il peut être intéressant de la conserver en mémoire sous forme d'une routine LM, et de loger dans un endroit relativement protégé de la mémoire vive système. On trouve justement de &479E à &47AF 18 octets (la longueur de la routine) correspondant au cinquième niveau des boucles FOR...NEXT. Si, comme moi, vous allez rarement au-delà de 3 niveaux d'imbricaiton, vous disposerez d'une commande "MERGE" résidente :
(1) CALL &479E ; (2) CLOAD "PROG" ; (3) POKE &46E1, 0, 56

Voici la routine désassemblée et son DUMP : MERGE

```
479E  10  LIDP   46E3
47A1  57  LDD
47A2  43  DECA
47A3  11  LIDL   E1
47A5  52  STD
47A6  11  LIDL   E4
47A8  57  LDD
47A9  2A  JRNCP  02 (47AC)
47AB  43  DECA
47AC  11  LIDL   E2
47AE  52  STD
47AF  37  RTN
```

```
4798  0000  0000  0000  1046
47A0  E357  4311  E152  11E4
47A8  572A  0243  11E2  5237
```

M. BALLAND

PROBABILITES

Nous n'avions pas encore publié de programme de ce type, cet oubli est aujourd'hui réparé. De plus, ces trois sous-routines regroupées sous un seul « titre » sont suffisamment simples pour être adaptées sans mal à tous les PC. A vous, ensuite, de les accomoder à votre goût.

EXPLICATIONS :

• Arrangements/Combinaisons : DEF V

Arrangements : probabilité d'obtenir un événement dans l'ordre, ex : le tiercé.
p = nombre d'éléments recherchés,
n = nombre d'éléments d'ensemble.

Tiercé de 20 chevaux dans l'ordre (et pour 5 francs...) :
p = 3, n = 20 donne A_{20}^{3} = 6840 on a donc 1 chance sur 6840 de gagner.
Combinaison : même principe mais l'événement n'a pas d'ordre.
Tiercé de 20 chevaux dans le désordre :
p = 3, n = 20 donne C_{20}^{3} = 1140 ; donc on a une chance sur 1140.
Conclusion idiote : c'est plus facile dans le...

• Loi binomiale : DEF B

n = nombre d'éléments de l'ensemble.
p = probabilité du choix favorable.
k = nombre d'essais.
Ex. pile ou face avec 20 pièces (ou la même lancée 20 fois).
n = 20, p = 0,5 car 50% face), k = 2 car 2 tirages.
Sur 2 tirages on a la probabilité d'avoir 2 fois face.
D'où p = 1,812E-4

• Loi de poisson : DEF H

λ = moyenne λ = n.p.
k = nombre de tirages.
La Loi de Poisson s'applique pour n très grand et p petit.

M. GIRBAL

```
4000:"** PROBABILITES *
     *
4010:"(c) M.GIRBAL & LE
     CLUB
5000:"V" WAIT 0
5002:CLS : PRINT "Arran
     /Comb"
5004:K$= INKEY$ : IF K$
     <>"A" AND K$<>"C"
     THEN 5004
5006:CLS : CURSOR 0,1:
     PRINT K$: CURSOR 1
     : INPUT "P",P
5008:CURSOR 49: INPUT "
     n",N
5010:A=1: FOR I=0 TO P-
     1
5012:A=A*(N-I): NEXT I
5014:IF K$="A" THEN 501
     8
5016:F=1: FOR I=1 TO P:
     F=F*I: NEXT I:A=A/
     F
5018:CURSOR 3,1: PRINT
     "=";A
5020:IF INKEY$ <>")"
     THEN 5020
5022:GOTO 5002
6000:"B" CLS : WAIT 0
6002:INPUT "n ?";N
6004:INPUT "P ?";P
6006:INPUT "k ?";K
6008:R=P^K*(1-P)^(N-K):
     A=1
6010:IF K=0 LET F=1:
     GOTO 6020
6012:FOR I=0 TO K-1
6014:A=A*(N-I): NEXT I
6016:F=1: FOR I=1 TO K
6018:F=F*I: NEXT I
6020:CLS : PRINT "P(x="
     ;K;")=";R*A/F
6022:K$= INKEY$ : IF K$
     ="B" THEN 6002
6024:IF K$<>")" THEN 60
     22
6026:GOTO 6006
7000:"H" WAIT 0: CLS
7002:GCURSOR (0,6):
     GPRINT "8464186080
     ": INPUT "?";L
7004:INPUT "K ?";K
7008:F=1: FOR I=1 TO K
7010:F=F*I: NEXT I
7012:P=L^K* EXP (-L)/F
7014:PRINT "P(x=";K;")=
     ";P
7016:GOTO 7004
```

ASTUCES ASTUCES



BOGUE EN ROM

L'instruction LINE (X1,Y1)-(X2,Y2),X,B devrait tracer un rectangle à l'écran, ayant son coin supérieur gauche aux coordonnées X1,Y1 et son coin inférieur droit en X2,Y2 et ceci en inversant la couleur des points formant le cadre. Or le point de coordonnées (X1,Y1) reste de la même couleur. L'explication est assez simple, faites ceci :
10:WAIT 0:CLS:LINE (0,0)-(149,31),X,B
20:GOTO 20
RUN (ENTER)
Vous verrez le coin supérieur gauche passer au noir, le tracé du rectangle continue, et en traçant le dernier côté il va trop loin d'un point puisqu'il réinverse le coin supérieur gauche ; donc il repasse dans sa couleur d'origine. Pour pallier cet inconvénient, il suffit de faire après l'instruction LINE « défectueuse » un LINE (X1,Y1)-(X1,Y1),X.



MODE ANGULAIRE

Il est difficile de savoir dans quel mode angulaire on se trouve. Je vous propose de mettre en réserve cette formule :
SHIFT MODE
SHIFT A
A:MID$(''DRG'',((PEEK &787C AND 7)/2)+1,1)
(ENTER)
MODE
et en faisant SHIFT A, la première lettre du mode angulaire dans lequel vous vous trouvez s'affiche.

M. GIRONDOT

PC 1350

DRIVER CLAVIER

Ce programme entièrement en L.M., non relogeable, est compris entre &6040 et &613E, il est donc nécessaire de réserver sa place dans la mémoire du 1350.

Pour cela, taper : POKE &6F01, &40, &61 suivi d'un NEW, en mode PRO. Vous pouvez alors entrer le programme par la méthode de votre choix.

Lancer le programme par CALL &6040 à chaque allumage de votre machine. Il est déconseillé de lancer, par la suite un autre programme par DEF « n », utilisez plutôt GOTO « n ».

Le programme est, en fait, une amélioration de l'interpréteur du PC ; il permet d'avoir accès à des fonctions prévues d'origine mais non exploitées. En effet, sous driver, un POKE &6F11, &80 produit l'autorepeat des touches ; un POKE &7886, &18 provoque un affichage vertical des données. La sonorisation est obtenue grâce aux codes ASCII des chiffres. La saisie s'effectue en &1204 et l'écriture en &A882. Le réglage des sons est obtenue en Pokant en &6108 une valeur correspondant à la longueur du son, la valeur d'origine étant 5.

En &6128, on règle la vitesse de répétition ; la valeur d'origine étant &10. Pour créer vous même vos propres fonctions, il vous suffira de modifier le JP de la table de sauts (&60C7-&60C8) par un JP de votre choix ; l'adresse d'origine étant &B06E correspondant à la routine de traitement de ENTER.

Ce programme a été écrit à l'aide de l'assembleur ASM 1350 actuellement commercialisé à la règle à calcul.

P. DEHLINGER

```
6040 78D90C78ACBBF204
6048 78610667182A032C
6050 08DA78A891796046
6058 4D4D4D4D4D4D4D4D
6060 4D4D4D4D4D670E2A
6068 15670A380F670438
6070 0D2A3367032805FE
6078 D00203F5D2F53BD4
6080 FC7A0860A06902DF
6088 6E03090104E27E0A
6090 04D80EE0AC0FE0EB
6098 13E1A314E197ACFA
60A0 F4EE796046670838
60A8 2EF53BD4FCF543D6
60B0 0229127A0560C969
60B8 05D5B206ACE507CF
60C0 140BE3D50CE465B0
60C8 6E2D2AF543D60229
60D0 2778AC4F2D3578D9
60D8 0C10783C57670828
60E0 0A02105278ACBB79
60E8 60466710280A0220
60F0 5278AC6479604602
60F8 085278AC76796046
6100 01DF2F016110DA02
6108 0534DA34125F6001
6110 DF2F016110DF2F0C
6118 34106F110280525B
6120 3467182A035B3702
6128 1034E9E82F031078
6130 8602184D10783CD4
6138 FE4D4D4D5B37AE
```

PC 1350 PC 2500

GESTION CE 516P

Voici un petit programme pour PC-1350 ou PC-2500 connecté via un CE-516L sur un CE-516P. Sans connaître les commandes du 516, faites des dessins, des textes,...

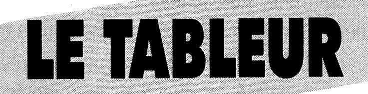
Pour utiliser ce programme, c'est très simple :
le mettre dans la mémoire de votre PC
faire RUN
taper la commande désirée, c'est à dire :
- un chiffre de 1 à 9 indiquant la direction dans laquelle vous voulez diriger le crayon. Sur la troisième ligne de l'écran, il est indiqué si le crayon est levé ou baissé (les directions sont représentées par une rose des vents centrée sur le 5 : 8=en haut 2=en bas...)
- la lettre M comme mode pour baisser ou lever la plume
- la lettre H comme Home pour revenir au centre de la feuille
- la commande L comme ligne pour indiquer le style de trait qui sera choisi ou entre ici une valeur allant de 0 à 15.
- la commande P comme impression pour écrire du texte. On indique la taille et le sens des caractères.
- la commande R comme Rond pour créer un cercle dont le centre est l'emplacement du stylo. On se déplace jusqu'au bord du cercle et on le fait tracer en appuyant sur +. On peut annuler cette commande en appuyant sur *
- la commande T comme teinture pour remplir un rond ou une boîte du motif saisi (valeur de 0 à 10)
- la commande C comme couleur pour changer de couleur. Entrer une valeur de 0 à 3
- et la commande F comme fin pour sortir du programme.

Bon dessin

```
10:OPEN "1200,N,8,1,A,L,&1A": CONSOLE
   160:ESC$= CHR$ &1B: DIM B$(0)*40
20:LPRINT ESC$+"@": BEEP 1: LPRINT ESC
   $+"b": LPRINT "M480,-662": LPRINT "
   I":P=0,Q$="dePl.",A$=".",X=0,Y=0
30:B$=A$: CLS : WAIT 0: CURSOR 17:
   PRINT "Ges516P": PRINT "Commande: _
   ": PRINT "Mode ";Q$
40:A$= INKEY$ : IF A$="" THEN 40
```

```
50:IF VAL A$<>0 CURSOR 34: PRINT A$:
   GOSUB 150: LPRINT CHR$ (77-9*(P=1))
   ;X;",";Y: GOTO 30
60:IF A$="m" OR A$="M" LET P=(P=0):
   GOTO 170
70:IF A$="h" OR A$="H" CURSOR 34:
   PRINT "Home": LPRINT "H": GOTO 30
80:IF A$="L" OR A$="l" CURSOR 34:
   PRINT "Ligne": GOTO 190
90:IF A$="P" OR A$="p" CURSOR 34:
   PRINT "impression": GOTO 210
100:IF A$="R" OR A$="r" CURSOR 34:
   PRINT "Rond": GOTO 260
110:IF (A$="T" OR A$="t") AND (B$="r"
   OR B$="R" OR B$="b" OR B$="B")
   CURSOR 34: PRINT "Teinture": GOTO 3
   30
120:IF A$="C" OR A$="c" CURSOR 34:
   PRINT "Couleur": GOTO 350
130:IF A$="B" OR A$="b" CURSOR 34:
   PRINT "Boite": GOTO 370
140:Z=40+400*((A$="F")+(A$="f")): BEEP
   1: GOTO Z
150:C= VAL A$-1:D= INT (C/3):E=C-D*3-1:
   D=D-1: IF ABS (X+10*E)>=480 OR ABS
   (Y+10*D)>=662 RETURN
160:X=X+10*E:Y=Y+10*D: RETURN
170:Q$-"depl.": IF P=1 LET Q$="ecrit."
180:GOTO 30
190:CURSOR 48: INPUT "Style de ligne:";
   L:L= INT ABS L: IF L<16 LPRINT "L";
   L
200:GOTO 30
210:CURSOR 48: INPUT "Taille des carac.
   :";C: CURSOR 72: INPUT "Direction:"
   ;D:C= INT ABS C:D= INT ABS D: IF C<
   64 AND D<4 THEN 230
220:GOTO 30
230:CURSOR 48: PRINT "
                                   ";:
   CURSOR 48
240:INPUT "texte:";B$(0): LPRINT ESC$+"
   ?";C: LPRINT "Q";D: LPRINT "P";B$(0
   ): LPRINT "M";X;",";Y
250:GOTO 30
260:CURSOR 48: PRINT "Allez au bord du
   rond.  (*) annuler  (+) tracer":V=X
   :W=Y
270:A$= INKEY$ : IF A$="" THEN 270
280:IF A$="*" LPRINT "M";V;",";W: GOTO
   320
290:IF A$="+" LET C= INT ( SQR ((V-X)^2
   +(W-Y)^2)+.5): LPRINT "M";V;",";W:
   LPRINT "C";V;",";W;",";C: GOTO 320
300:IF VAL A$=0 THEN 270
310:GOSUB 150: LPRINT "M";X;",";Y: GOTO
   270
320:A$= CHR$ (66+(A$="*")):X=V:Y=W:
   GOTO 30
330:CURSOR 48: INPUT "Style de rempliss
   age:";C:C= INT ABS C: IF C<11
   LPRINT "T";C
340:GOTO 30
350:CURSOR 48: INPUT "0:noir 1:bleu 2:v
   ert    3:rouge ?";C:C= INT ABS C:
   IF C<4 LPRINT ESC$;C
360:GOTO 30
370:CURSOR 48: PRINT "Allez a l'autre b
   out.   (*) annuler  (+) tracer":V=X
   :W=Y
380:A$= INKEY$ : IF A$="" THEN 380
390:IF A$="*" LPRINT "M";V;",";W: GOTO
   430
400:IF A$="+" LPRINT "D";V;",";W;",B":A
   $="B": GOTO 430
410:IF VAL A$=0 THEN 380
420:GOSUB 150: LPRINT "M";X;",";Y: GOTO
   380
430:X=V:Y=W: GOTO 30
440:CLS : PRINT "Au Revoir ...": CLOSE
   : END
```

PC2500



Une astuce...

Lorsque vous modifiez des éléments du tableau, vous pouvez faire des modifications sur ce qui a déjà été introduit dans une cellule.

Tapez une touche puis vous accédez à ce moment à une cellule dont vous pouvez modifier le contenu.

Des formules...

Une petite gestion de stock.

« produits » : Total-HT=qte+PU-HT : Total TTC=Total-HT*(1+TVA/100)

Gérez vos actions et vos obligations.

« Action » : Emission : Total=Actuel*Qte : Différence=Total-Emission*Qte

« Obligation » : Actuel=P%/100*Emission : Total=Actuel*Qte : Gain=Total-Emission*Qte

Une autre astuce...

Un INKEY$ sans test. Au lieu de faire :

10 A$=INKEY$: IF A$=" " THEN 10

Tapez :

10 CALL&802D : A$=CHR$PEEK&6D89

Vous gagnerez du temps et de la place mémoire.

CODES INCONNUS DU SC 61860

LM SC 61860 (ESR-H)

Comme en témoigne votre courrier, vous n'avez pas été sans remarquer que certaines instructions du microprocesseur SC 61860 équipant à ce jour la plupart des PC, figurent dans notre tableau publié dans la revue N° 8, mais ne sont pas expliquées dans le « MACHINE LANGUAGE REFERENCE MANUAL » du PC 1250/51. Voici donc pour les mordus de la « chose », les fonctions connues de ces instructions « oubliées ». Elles sont présentées à la manière du manuel et plusieurs codes nous sont encore inconnus. Donc à vous de jouer.

J.F. V.

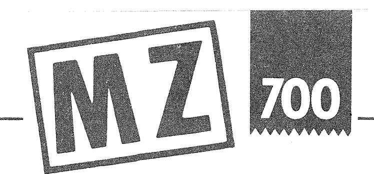
d est défini pour IF GTO ELSE, auparavant par ON RET ; ces deux instructions étant couplées. L'adresse après RET est l'adresse de retour (après rencontre de RTN & 37) d'une routine appelée par GTO ou ELSE.
N est le registre 0A de la pile, O est le registre 0B de la pile tout comme A est le registre 02 de cette même pile.

Mmémonique	Fonction	Code	Octets
DATA	I ⟶ d, BA ⟶ nm REPEAT B, A ⟶ (P), nm + 1, d − 1 UNTIL d = FF B, A et I inchangés P changé	35	1
READ	(PC + 1) ⟶ A	56	1
READM	(PC + 1) ⟶ (P)	54	1
TSMA	(P)˜A Z	C6	1
INCN	N + 1 ⟶ N	4A	1
DECN	N − 1 ⟶ N	4B	1
INCO	0 + 1 ⟶ 0	CA	1
DECO	0 − 1 ⟶ 0	CB	1
ON k RETnm	k ⟶ d nm ⟶ (R − 1, R − 2)	7A	4
IF a_d GTO $n_d m_d$ a_{d-1} GTO n_{d-1} m_{d-1} a_1 GTO $n_1 m_1$ ELSE nm	d REPEAT A − a_d, d − 1 Si Z = 1 alors n_d ⟶ PCH m_d ⟶ PCL UNTIL d = FF ou Z = 1 Sinon n ⟶ PCH m ⟶ PCL	143d + 3	

POINTEURS SYSTEMES DES PC 12XX/14XX/13XX

Ce tableau est le premier d'une série comportant les adresses des principaux pointeurs systèmes des PC équipés du microprocesseur ESR-H. Ceci permettra, nous l'espérons, des adaptations plus aisées de nos programmes y faisant référence, pour les différentes machines citées.

DEFINITION	ADRESSES EN & SUR PC 1245/55/51	ADRESSES EN & SUR PC 1401/02	ADRESSES EN & SUR PC 1260/61	ADRESSES EN & SUR PC 1350
START BASIC (PDP)	C6E1, C6E2	46E1, 46E2	66E1, 66E2	6F01, 6F02
END BASIC (PFP)	C6E3, 66E4	46E3, 46E4	66E3, 66E4	6F03, 6F04
WARTPR (PDV)	C6FC, C6FD	46FC, 46FD	66FC, 66FD	6F07, 6F08
OPERANDE DE WAIT	C6E5, C6E6	46E5, 46E6	66E5, 66E6	70B3, 70B4
START (RSV-1)	C6C4, C6C5	/	283E, 283F	6F55, 6F56
79 JP n m	C6DB	46DB	66DB	6F59
n	6CDC	46DC	66DC	6F5A
m	C6DD	46DD	66DD	6F5B
Flag WAIT n	C6D8	46D8	66D8	6F13
Flag P. = <P.	C6D7	46D7	66D7	6F12
VARIABLES FIXES A Z	C5DO C69F	45DO 469F	6500 65CF	6C30 6D00
RSV	B800 B82F sauf 1245	/	65D0 65FF	6E6F 6FFF



TORTUE GRAPHIQUE SUR IMPRIMANTE MZ-700

Ce petit programme permet de s'initier à la partie graphique du langage LOGO sur l'imprimante du MZ-700. Il est écrit en version 4 (au moins) du K-Basic V. 5 et occupe environ 11K octets en mémoire.

Le dessin est toujours tracé dans un carré de 480 unités sur 480, le centre étant placé au centre de la bande de papier de l'imprimante. On ne peut pas sortir de ce cadre.

Le programme possède un certain nombre de commandes que nous allons étudier successivement. Auparavant, précisons que sur l'écran, seront inscrites les valeurs des variables X, Y (position stylo), S (variable SCALE qui sera vue plus loin) et A, angle que fait la direction du stylo avec l'axe Ox. Cette direction est matérialisée par le pointeur se trouvant dans le carré rouge de l'écran.

CENTRE :
Ramène le stylo au centre sans tracer de trait. Remet X et Y à zéro ainsi que l'angle A. Ne modifie pas S.

CHANGES :
Avances le papier et réinitialise la tortue. Cet ordre correspons à CLS et permet de changer de dessin.

LEVES :
Permet de lever le stylo et donc de ne tracer aucun trait lors de ses déplacements.

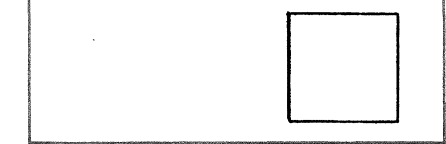
BAISSES :
Permet au stylo de laisser une trace sur le papier. Au départ, le stylo est considéré comme baissé.

DROITE :
DROITE argument fait tourner le stylo sur la droite, d'un angle égal à la valeur de « argument » (Le programme retranche « argument » à A).

GAUCHE :
Comme DROITE mais en sens inverse. On remarquera que pour que le programme comprenne bien l'ordre, il faut laisser au moins un espace entre DROITE (ou GAUCHE) et argument. On pourra essayer plusieurs exemples et voir comment se modifie le pointeur sur l'écran, ainsi que A : DROITE 56 + 42

AVANCES :
AVANCES argument fait avancer le stylo de la longueur « argument » dans la direction indiquée par A.

RECULES :
RECULES argument, le fait reculer. Par exemple donner 4 fois de suite les ordres : AVANCES 100 et DROITE 90 :



VAS-EN :
VAS-EN X Y déplace le stylo au point X, Y sur le papier. X et Y peuvent être des expressions que le Basic calculera.

COULEUR :
COULEUR n où est une expression numérique permet de choisir la couleur du stylo. On peut d'ailleurs écrire la couleur en toutes lettres :
COULEUR 0 ou COULEUR NOIR
COULEUR 1 ou COULEUR BLEU
COULEUR 2 ou COULEUR VERT
COULEUR 3 ou COULEUR ROUGE

FINI :
Interrompt le programme. Le retour se fait à la ligne 50.

POUR :
POUR permet de définir de nouvelles commandes qui seront incorporées au « mini-langage ». Il est suivi du nom donné à cette nouvelle commande. Par exemple, pour définir un carré de 100 unités de côté, on fera :

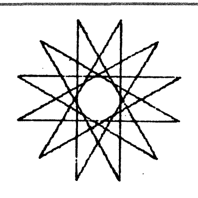
```
POUR CARRE
AVANCES 100
DROITE 90
AVANCES 100
DROITE 90
AVANCES 100
DROITE 90
AVANCES 100
DROITE 90
FIN
```

Pour l'utiliser, il suffira d'écrire : CARRE

REMARQUES : Il n'est pas possible de définir une commande portant le même nom qu'une commande déjà définie.
Il est possible de lister sur l'écran CARRE (par exemple) en faisant LISTES CARRE.
De même, le listing sur l'imprimante est possible par PLISTES CARRE.
On peut supprimer CARRE de la mémoire en écrivant OUBLIES CARRE.
Enfin, il est possible de modifier une commande de CARRE par EDITES CARRE. L'écran est alors changé et trois options (effacement, modification ou insertion de lignes) avec, bien entendu FIN permettent de transformer la commande.

REPETES :
REPETES argument : Cette structure effectue les instructions...
... N fois, N étant la valeur de « argument ».
ENCORE

```
POUR ETOILE
REPETES 12
AVANCES 150
DROITE 150
ENCORE
FIN
```



SAUVES :
SAUVES argument enregistre sur la cassette, toutes les nouvelles commandes déjà définies, sous le nom de fichier « argument ».

CHARGES :
CHARGES argument replace en mémoire, venant de la cassette, les définitions enregistrées sous le nom de fichier « argument ».

ECHELLE :

ECHELLE argument donne à la variable SCALE (apparaissant à l'écran sous le nom S) la valeur de argument. Son effet est de changer les déplacements relatifs du stylo.

Avec ECHELLE 2 AVANCES 50 fait avancer le stylo de 50 unités (pour l'imprimante du MZ-700)

Avec ECHELLE 2 AVANCES 50 provoquera une avance 100 unités.

TAILLE :

TAILLE argument multiplie SCALE par la valeur de l'argument.

Par exemple TAILLE 0.5 puis ETOILE dessinera une étoile deux fois plus petite.

LIGNE :

LIGNE argument est un entier de 1 à 16 permet de modifier le type de ligne tracée par le stylo (Voir page 87 manuel de l'ordinateur).

SOIT :

SOIT permet au programme d'utiliser des variables. SOIT COTE 15 + 7 donnera à la variable COTE la valeur 22. On remarquera que COTE est alors une variable considérée comme celles du programme, par le Basic. Autrement dit, il ne faudra pas utiliser comme variable une des variables du programme lui même, sous peine de le voir fonctionner de façon aberrante.

SI. SINON :

C'est le IF THEN ELSE du Basic : SI condition action I SINON action 2. Il peut être utile pour la condition, d'utiliser certaines fonctions du Basic ce qui peut poser des problèmes aux programmes (pour lequel, un espace est un séparateur entre diverses commandes). On pourra alors écrire la condition entre crochets : (cf ligne 3).

```
POUR DOUBLEPOLY
Parametres: REP C0 C1 A1 F1
 1  SOIT KA A
 2  REPETES REP
 3  SI [(KA MODULO(F1*A1)]
(F1*A1/2)] AVA
NCES C0 SINON AVANCES C1
 4  DROITE A1
 5  SOIT KA KA+A1
 6  ENCORE
 7  FIN
```

UTILISATION DE PARAMETRES :

L'exemple précédent montre qu'il est possible de définir de nouvelles commandes possédant des paramètres. Il peut y avoir au maximum 5 paramètres (exemple : On entre POUR DOUBLEPOLY REP CO CI AI FI) et lors de l'appel on indique les valeurs : DOUBLEPOLY 50 50 100 30 120. Le programme affectera à REP la valeur 50, à CO la valeur 50,..., à FI la valeur 120. A la fin, REP, CO,..., FI retrouveront leurs valeurs intiales. Il y a donc, comme en Basic, passage de paramètres, par valeur.

```
POUR CARRE
Parametres: COTE
 1  REPETES 4
 2  AVANCES COTE
 3  DROITE 90
 4  ENCORE
 5  FIN

POUR CARRES

 1  SOIT COTE 0
 2  REPETES 20
 3  SOIT COTE COTE+5
 4  CARRE COTE
 5  ENCORE
 6  FIN
```

REMARQUES :

Shift F5 permet de réinitialiser l'imprimante (en cas d'erreur).

Shift F4 permet de faire redémarrer le programme après un Break, sans perdre le contenu de la mémoire.

Les valeurs X et Y affichées par le programme, en haut à gauche de l'écran sont les parties entières de X et Y, positions du stylo. Elles sont donc « arrondies » par rapport aux vraies valeurs. D'autre part, il se produira des erreurs sur X et Y car les variables sont calculées par des Sinus et Cosinus d'angles exprimés en degrés.

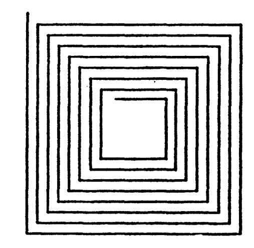
QUELQUES EXEMPLES

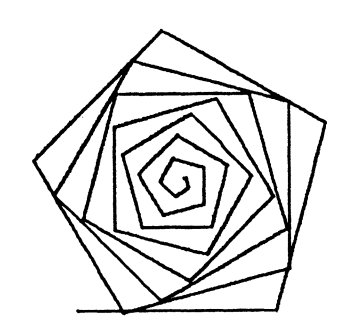
Autre définition d'un carré. Il faudra faire Shift Break pour arrêter le programme et le reprendre par Shift F4

```
POUR CARRE

 1  AVANCES 100
 2  DROITE 90
 3  CARRE
 4  FIN
```

```
POUR MAZE
Parametres: DISTANCE
 1  AVANCES DISTANCE
 2  DROITE 90
 3  MAZE DISTANCE+5
 4  FIN
```

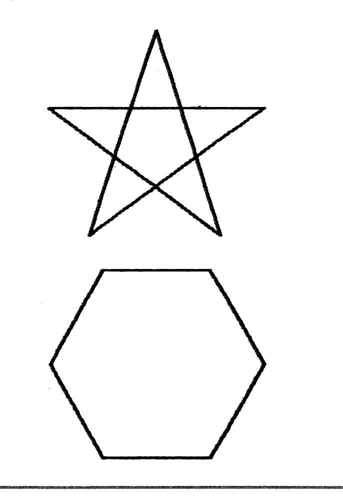




```
POUR SPIRAL
Parametres: DISTANCE
 1  AVANCES DISTANCE
 2  DROITE 75
 3  SPIRAL DISTANCE+5
 4  FIN
```

On fera attention au fait que ces deux derniers types de programme s'appellent eux mêmes et par suite, si on les laisse tourner trop longtemps, ils satureront la mémoire du MZ-700 ce qui détruira toutes les variables.

```
POUR POLYGONE
Parametres: COTE ANGLE
 1  AVANCES COTE
 2  DROITE ANGLE
 3  POLYGONE COTE ANGLE
 4  FIN
```

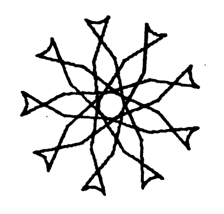


```
POUR CORPS

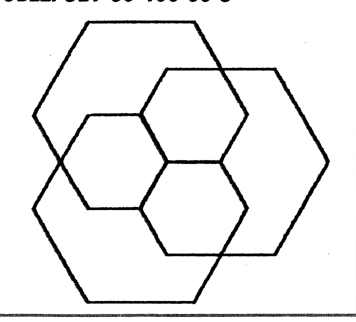
 1  SI [[KA MODULO(F1*A1)]
<F1*A1/2] AVAN
CES C0 SINON AVANCES C1
 2  DROITE A1
 3  SOIT KA KA+A1
 4  CORPS
 5  FIN

POUR DOUBLEPOLY
Parametres: C0 C1 A1 F1
 1  SOIT KA 0
 2  CENTRE
 3  CORPS
 4  FIN
```

DOUBLEPOLY 45-6 25 8



DOUBLEPOLY 50 100 60 8

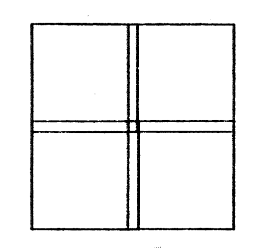


IMPOLY 100 10 90 5

```
POUR IMPOL

 1  SI [[KA MODULO
(F1*A1)]=0] AVANCES S
2 SINON AVANCES S1
 2  DROITE A1
 3  SOIT KA KA+A1
 4  IMPOL
 5  FIN

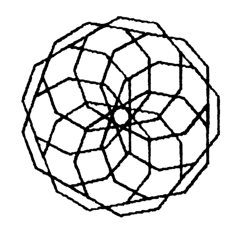
POUR IMPOLY
Parametres: S1 S2 A1 F1
 1  CENTRE
 2  SOIT KA 0
 3  IMPOL
 4  FIN
```



UNE AUTRE DEFINITION :

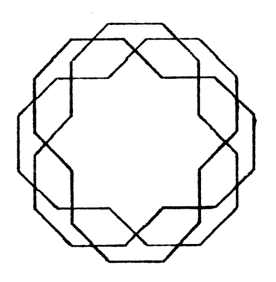
IMPOLY 35-20 40 25

```
POUR IMPOLY
Parametres: S1 S2 A1 F1
 1  CENTRE
 2  SOIT KA 0
 3  REPETES 500
 4  SI [[KA MODULO
(F1*A1)]=0] AVANCES S
2 SINON AVANCES S1
 5  DROITE A1
 6  SOIT KA KA+A1
 7  ENCORE
 8  FIN
```

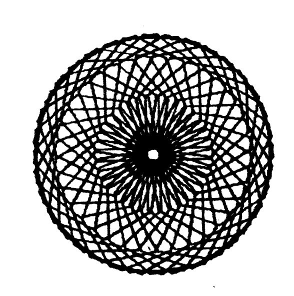


```
POUR AIMPOLY
Parametres: S1 A1 A2 F
 1  VAS-EN -100 0
 2  SOIT KA 0
 3  REPETES 500
 4  AVANCES S1
 5  SI [[KA MODULO F]=0]
DROITE A1 SINON
DROITE A2
 6  SOIT KA KA+1
 7  ENCORE
 8  FIN
```

AIMPOLY 50-45 45 5



AIMPOLY 50 50-50 7



VARIABLES UTILISEES PAR LE PROGRAMME

A	Angle entre l'axe des abscisses et la direction du stylo.
A§	Utilisée par le programme
BLEU	= 1 (Couleur du stylo)
B§	Utilisé par le programme
CO	= Cos(A)
C	
C§	
D	
DUMMY	Variable utilisée pour le passage de variables entre le Basic et le programme lui-même. Il ne faudra surtout pas changer les deux premières lignes (10 et 20) du programme.
E	
E1	
FL	Indique le type de ligne tracée par le stylo
I	
J	
KS	= si stylo baissé et 0 sinon
KW	= 25 nombres d'instructions prédéfinies
K	
L	
NBW	= degré de remplissage du tableau C§ contenant les instructions prédéfinies et celles définies par l'utilisateur.
NOIR	= (Couleur du stylo)
N	
P	
ROUGE	= 3 (Couleur du stylo)
R§	
SI	= Sinus(A)
SCALE	Echelle
T	
T1	
VERT	= 2 (Couleur du stylo)
X	Abscisse du stylo
XO	
Y	Ordonnée du stylo
YO	
Z	
Tableaus W§, C§, P§, P, Q, ER§	

B. KOKANOSKI

UNE AMELIORATION DU K-BASIC V.5.4

Nous allons ajouter au Basic une possibilité très intéressante, celle de pouvoir nous donner, sur l'écran ou sur imprimante, le nom et la valeur des variables présentes en mémoire, ainsi que le nom et le numéro de ligne des définitions de procédures et de fonctions.

CTRL Z donnera sur écran, le listing des FN / PROC contenues dans le programme, avec le numéro de la ligne où elles se trouvent. La routine qui va être ajoutée au Basic, lit tout le programme (et non le catalogue des FN/PROC) et, par suite, fonctionnera même si le programme n'a pas encore été exécuté.

CTRL C fait la même chose, mais sur imprimante. Il ne se passera rien si l'imprimante est en mode graphique.

CTRL J donne sur écran, les noms et valeurs des diverses variables utilisées par le programme (celui-ci doit avoir déjà fonctionné).

CTRL L fait la même chose mais sur imprimante. Il ne se passera rien si l'imprimante est en mode graphique.

Remarque :
Pour toutes ces routines, le listing est stoppé momentanément, par appui sur la barre d'espace et définitivement par Shift Break.

Pour implanter ces possibilités dans K-Basic V.5.4, le charger et par Shift Break retourner au moniteur en mémoire morte.

Par M modifier les octets IOFD et y écrire 35 (identificateur K-Basic V.5.5)

IOFD	35
126D	60
126E	6D
127B	77
127C	6D
127F	65
1280	6D
129B	72
129C	6D

Toujours par M introduire, à partir de l'octet 7E52 (= 6C52 + 1 200) les codes du listing suivant : 4F 06 00 .... 20 DC C9.
Enfin, par M, introduire à partir de l'octet 7F60 (= 6D60 + 1 200) les codes du second listing : 21 38 6E .... 00 20 46 4E 00.

On réenregistrera le Basic par J 8670 et enfoncement de S.
Après toutes ces modifications, les zones libres du Basic sont :
6AF2 à 6C5I – 6D4I à 6D5F – 6FF3 à 6FFC

B.Kokanoski

```
0000  P              NL:      EQU   0009H
0000  P              PRINTM:  EQU   1819H
0000  P              DEBVAR:  EQU   73A3H
0000  P              WRIT:    EQU   17D3H
0000  P              CHINCC:  EQU   4D62H
0000  P              SAUTLN:  EQU   17F9H
0000  P              #FAC1:   EQU   72A1H
0000  P              NUMPRT:  EQU   35A3H
0000  P              ECRMES:  EQU   17E2H
0000  P              GTKEY:   EQU   0058H
0000  P              BLOCKM:  EQU   4885H
0000  P              DEBPRO:  EQU   73A1H
0000  P              CHLASC:  EQU   21DBH
0000  P              SPAC0:   EQU   4F41H
0000
0000                          REL   6C52H
6C52
6C52 4F              TABLO:   LD    C,A
6C53 0600                     LD    B,00H
6C55 09                       ADD   HL,BC
6C56 7E                       LD    A,(HL)
6C57 32956C                   LD    (NBDIM),A
6C5A 47                       LD    B,A
6C5B ED5BA173                 LD    DE,(DEBPRO)
6C5F 1B                       DEC   DE
6C60 ED53976C                 LD    (BUFDEB),DE
6C64 AF                       XOR   A
6C65 12              S7:  --> LD    (DE),A
6C66 1B                       DEC   DE
6C67 23                       INC   HL
6C68 12                       LD    (DE),A
6C69 1B                       DEC   DE
6C6A 23                       INC   HL
6C6B 10F8                     DJNZ  S7
6C6D 22BC6C                   LD    (PTDIM),HL
6C70 23                       INC   HL
6C71 23              TABL1:-->INC   HL
6C72 23                       INC   HL
6C73 22F16D                   LD    (POINTR),HL
6C76 CDD86D                   CALL  LISTE1
6C79 3E28                     LD    A,28H
6C7B CDD317                   CALL  WRIT
6C7E CD946C                   CALL  PARA
6C81 F5                       PUSH  AF
6C82 CDF06D                   CALL  VARSP
6C85 2AF16D                   LD    HL,(POINTR)
6C88 23                       INC   HL
6C89 F1                       POP   AF
6C8A CAD56D                   JP    Z,TABLO1
6C8D CD896D                   CALL  ATTEND
6C90 20DF                     JR    NZ,TABL1
6C92 E1                       POP   HL
6C93 C9                       RET
6C94 06              PARA:    DEFB  06H
6C95 00              NBDIM:   DEFB  00H
6C96 21                       DEFB  21H
6C97 0000            BUFDEB:  DEFW  0000H
6C99 56              S5:  --> LD    D,(HL)
6C9A 2B                       DEC   HL
6C9B 5E                       LD    E,(HL)
6C9C 2B                       DEC   HL
```

BC = longueur du nom de la variable.

Saute le nom en zone I
A = nombre de dimensions. Le stocke dans la routine elle-même.
B = nb de dimensions.
Va stocker (en remontant) B fois 0000 dans le Buffer situé juste avant le début du programme utilisateur.

Fait ce stokage. C'est dans cette zone que la routine trouvera les nombres à écrire (le dernier étant le premier indice...)
Profite de la boucle, pour sauter, en Zone I, les dimensions.
Pointe sur octet haut 1ère dim (cf NOTICE)
Saute la « longueur des zones Z »

Pointe la zone de 3 octets : L n° d'ordre
Ecrit le nom de la variable
Ecrit (

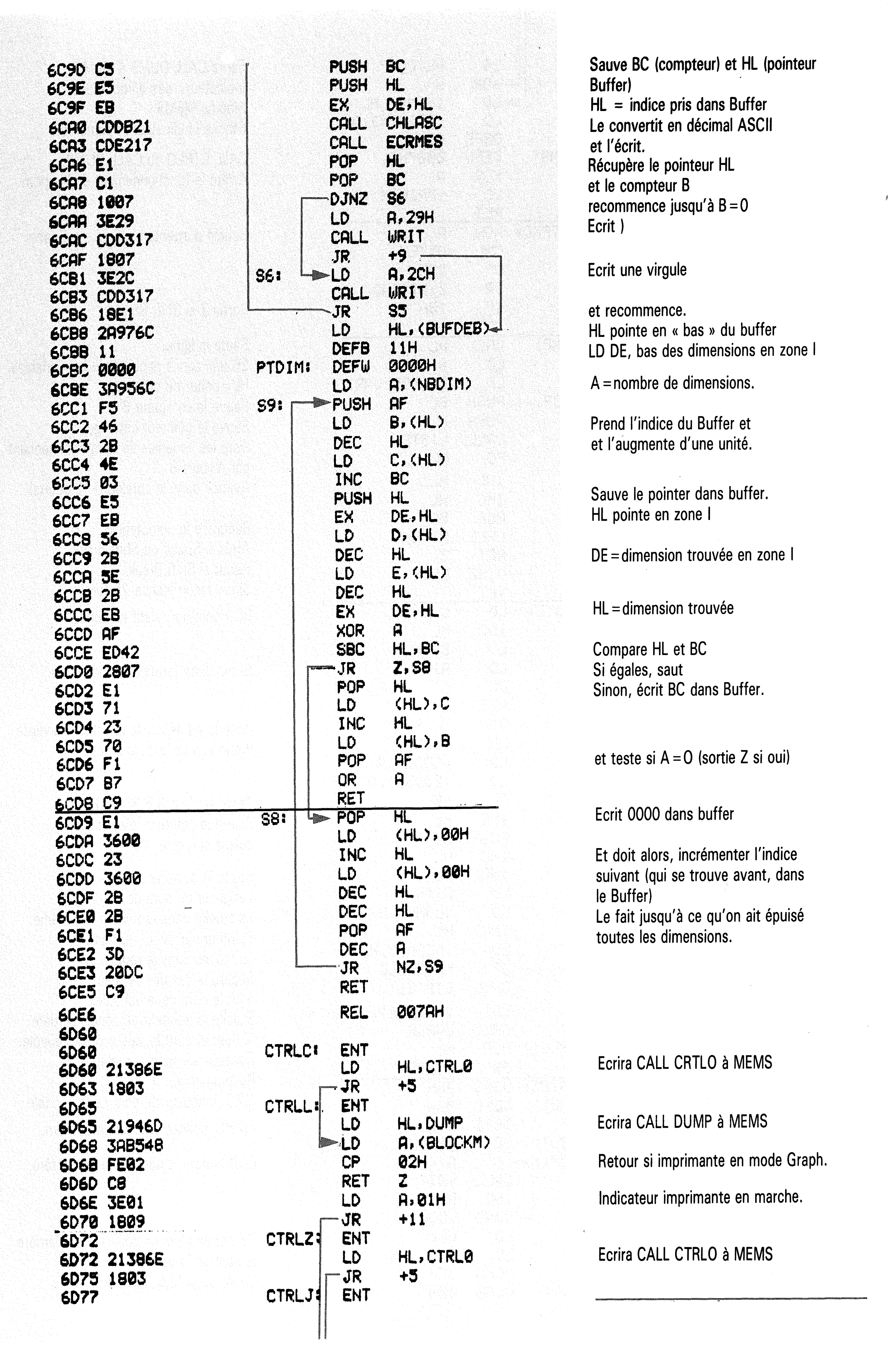
Ecrit les indices Z si terminé.
Sauve Z ou NZ
Ecrit la valeur
Avance dans la cellule en zone I

Si tableau épuisé, sortie
Teste Shift Break et Space
Recommence pour l'élément suivant
Retour si Shift Break.

LD B, nombre de dimensions.

LD HL, fin de buffer où on a mis les 0000

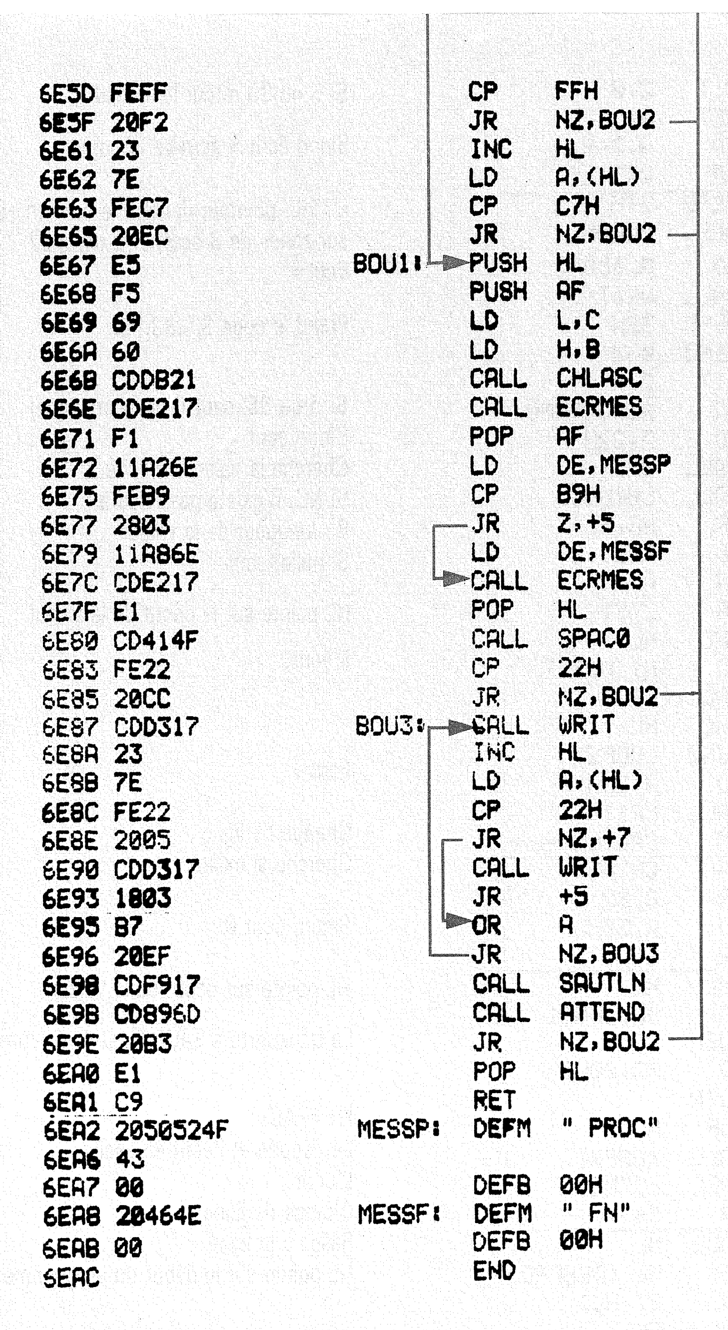
Prend dans DE le contenu (en remontant dans la mémoire).



```
6C9D C5              PUSH  BC             Sauve BC (compteur) et HL (pointeur
6C9E E5              PUSH  HL             Buffer)
6C9F EB              EX    DE,HL          HL = indice pris dans Buffer
6CA0 CDDB21          CALL  CHLASC         Le convertit en décimal ASCII
6CA3 CDE217          CALL  ECRMES         et l'écrit.
6CA6 E1              POP   HL             Récupère le pointeur HL
6CA7 C1              POP   BC             et le compteur B
6CA8 1007            DJNZ  S6             recommence jusqu'à B=0
6CAA 3E29            LD    A,29H          Ecrit )
6CAC CDD317          CALL  WRIT
6CAF 1807            JR    +9
6CB1 3E2C      S6:   LD    A,2CH          Ecrit une virgule
6CB3 CDD317          CALL  WRIT
6CB6 18E1            JR    S5             et recommence.
6CB8 2A976C          LD    HL,(BUFDEB)    HL pointe en « bas » du buffer
6CBB 11              DEFB  11H            LD DE, bas des dimensions en zone I
6CBC 0000      PTDIM: DEFW 0000H
6CBE 3A956C          LD    A,(NBDIM)      A=nombre de dimensions.
6CC1 F5        S9:   PUSH  AF
6CC2 46              LD    B,(HL)         Prend l'indice du Buffer et
6CC3 2B              DEC   HL             et l'augmente d'une unité.
6CC4 4E              LD    C,(HL)
6CC5 03              INC   BC
6CC6 E5              PUSH  HL             Sauve le pointer dans buffer.
6CC7 EB              EX    DE,HL          HL pointe en zone I
6CC8 56              LD    D,(HL)
6CC9 2B              DEC   HL             DE=dimension trouvée en zone I
6CCA 5E              LD    E,(HL)
6CCB 2B              DEC   HL
6CCC EB              EX    DE,HL          HL=dimension trouvée
6CCD AF              XOR   A
6CCE ED42            SBC   HL,BC          Compare HL et BC
6CD0 2807            JR    Z,S8           Si égales, saut
6CD2 E1              POP   HL             Sinon, écrit BC dans Buffer.
6CD3 71              LD    (HL),C
6CD4 23              INC   HL
6CD5 70              LD    (HL),B
6CD6 F1              POP   AF             et teste si A=0 (sortie Z si oui)
6CD7 B7              OR    A
6CD8 C9              RET
6CD9 E1        S8:   POP   HL             Ecrit 0000 dans buffer
6CDA 3600            LD    (HL),00H
6CDC 23              INC   HL             Et doit alors, incrémenter l'indice
6CDD 3600            LD    (HL),00H       suivant (qui se trouve avant, dans
6CDF 2B              DEC   HL             le Buffer)
6CE0 2B              DEC   HL             Le fait jusqu'à ce qu'on ait épuisé
6CE1 F1              POP   AF             toutes les dimensions.
6CE2 3D              DEC   A
6CE3 20DC            JR    NZ,S9
6CE5 C9              RET
6CE6                 REL   007AH
6D60
6D60           CTRLC: ENT
6D60 21386E          LD    HL,CTRL0       Ecrira CALL CRTLO à MEMS
6D63 1803            JR    +5
6D65           CTRLL: ENT
6D65 21946D          LD    HL,DUMP        Ecrira CALL DUMP à MEMS
6D68 3AB548          LD    A,(BLOCKM)
6D6B FE02            CP    02H            Retour si imprimante en mode Graph.
6D6D C8              RET   Z
6D6E 3E01            LD    A,01H          Indicateur imprimante en marche.
6D70 1809            JR    +11
6D72           CTRLZ: ENT
6D72 21386E          LD    HL,CTRL0       Ecrira CALL CTRLO à MEMS
6D75 1803            JR    +5
6D77           CTRLJ: ENT
```

```
6D77 21946D                LD    HL,DUMP        Ecrira CALL DUMP à MEMS
6D7A AF                    XOR   A              Indicateur : pas d'imprimante.
6D7B 22826D                LD    (MEMS),HL      Modifie MEMS
6D7E 321918                LD    (PRINTM),A     Stocke l'indicateur imprimante.
6D81 CD                    DEFB  CDH
6D82 0000         MEMS:    DEFW  0000H          CALL CTRLO ou CALL DUMP.
6D84 AF                    XOR   A              Arrête le fonctionnement de l'imprim.
6D85 321918                LD    (PRINTM),A
6D88 C9                    RET
----------------------------------------------------------------------
6D89 AF           ATTEND:  XOR   A
6D8A CD5800                CALL  GTKEY          Boucle d'attente si appui sur Space.
6D8D FE20                  CP    20H
6D8F 28F8                  JR    Z,ATTEND
6D91 FE1B                  CP    1BH
6D93 C9                    RET                  Sortie Z si Shift Break.
----------------------------------------------------------------------
6D94 CD0900       DUMP:    CALL  NL             Saute la ligne.
6D97 061A                  LD    B,1AH          26 (IA) zones dans catalogue variables.
6D99 2AA373                LD    HL,(DEBVAR)    HL=début du catalogue.
6D9C C5           LOOP:    PUSH  BC             Sauve le compteur B
6D9D E5                    PUSH  HL             Sauve le pointeur catalogue.
6D9E CDAC6D                CALL  LISTE          Liste les variables de nom commençant
                                                par A, puis B,...
6DA1 E1                    POP   HL
6DA2 23                    INC   HL             Avance dans le catalogue (2 octets)
6DA3 23                    INC   HL
6DA4 C1                    POP   BC             Récupère le compteur B
6DA5 CD896D                CALL  ATTEND         Arrêt si Space ou Shift Break
6DA8 C8                    RET   Z              retour si Shift Break
6DA9 10F1                  DJNZ  LOOP           Sinon recommence 26 fois.
6DAB C9                    RET
----------------------------------------------------------------------
6DAC 4E           LISTE:   LD    C,(HL)         BC=pointeur relatif en zone I
6DAD 23                    INC   HL
6DAE 46                    LD    B,(HL)
6DAF 78                    LD    A,B
6DB0 B1                    OR    C              Si nul, liste terminée donc retour.
6DB1 C8                    RET   Z
6DB2 09                    ADD   HL,BC          Accède au début de la cellule suivante
6DB3 7E                    LD    A,(HL)         Prend son code et le stocke.
6DB4 32E76D                LD    (CODE),A
6DB7 32F96D                LD    (CODE1),A
6DBA FE40                  CP    40H            Teste si c'est 3,5,83 ou 85
6DBC 23                    INC   HL             Pointe le pointeur relatif de la
6DBD E5                    PUSH  HL             cellule suivante.
6DBE 23                    INC   HL
6DBF 23                    INC   HL             Saute le pointeur relatif
6DC0 7E                    LD    A,(HL)         Longueur du nom de la variable
6DC1 32D96D                LD    (LONG),A       Le stocke dans la routine elle-même.
6DC4 23                    INC   HL             Pointeur sur début du nom.
6DC5 22DB6D                LD    (DEBUT),HL     Le stocke dans la routine.
6DC8 D2526C                JP    NC,TABLO       Sépare le cas des tableaux
6DCB CDD86D                CALL  LISTE1         écrit le nom de la variable.
6DCE ED53F16D              LD    (POINTR),DE    Stocke le pointeur en zone I derrière
6DD2 CDF06D                CALL  VARSP          le nom et écrit la valeur de la variable.
6DD5 E1           TABLO1:  POP   HL             Pointeur sur pointeur relatif
6DD6 18D4                  JR    LISTE          Recommence.
----------------------------------------------------------------------
6DD8 06           LISTE1:  DEFB  06H            LD B, longueur du nom de la variable.
6DD9 00           LONG:    DEFB  00H
6DDA 21                    DEFB  21H            LD HL, pointeur sur début du nom.
6DDB 0000         DEBUT:   DEFW  0000H
6DDD 7E           LOOP1:   LD    A,(HL)         Ecrit le nom, caractère par caractère.
6DDE CDD317                CALL  WRIT
6DE1 23                    INC   HL
6DE2 10F9                  DJNZ  LOOP1
6DE4 5D                    LD    E,L            DE pointe alors en zone I juste derrière
6DE5 54                    LD    D,H            le nom de la variable.
6DE6 3E                    DEFB  3EH            LD A, code (3,5,83 ou 85)
6DE7 00           CODE:    DEFB  00H
```

```
6DE8 CB57              BIT   2,A            Si 5 ou 85 retour immédiat.
6DEA C0                RET   NZ
6DEB 3E24              LD    A,24H          Sinon écrit $ derrière le nom.
6DED C3D317            JP    WRIT
6DF0 21       VARSP:   DEFB  21H            LD HL, pointeur derrière le nom en zone I
6DF1 0000     POINTR:  DEFW  0000H          sur zones de 3 octets : L ordre Z2
6DF3 3E3D              LD    A,3DH          Ecrit =
6DF5 CDD317            CALL  WRIT
6DF8 3E                DEFB  3EH            Prend le code 3,5,83,85
6DF9 00       CODE1:   DEFB  00H
6DFA CB57              BIT   2,A
6DFC 201E              JR    NZ,VARNMB      Si 5 ou 85, saut (Var. Numérique)
6DFE 3E22              LD    A,22H          Sinon écrit «
6E00 CDD317            CALL  WRIT
6E03 CD624D            CALL  CHINDC         Cherche la valeur en zone 2
6E06 300C              JR    NC,S1          Si NC, n'existe pas encore
6E08 78                LD    A,B            B=longueur de la valeur
6E09 B7                OR    A              Si nulle, saut
6E0A 2808              JR    Z,S1
6E0C 23                INC   HL             HL pointe sur le début de la valeur
6E0D 7E       LOOP2:   LD    A,(HL)         L'écrit.
6E0E CDD317            CALL  WRIT
6E11 23                INC   HL
6E12 10F9              DJNZ  LOOP2
6E14 3E22     S1:      LD    A,22H          Ecrit «
6E16 CDD317   S3:      CALL  WRIT
6E19 C3F917   S4:      JP    SAUTLN         Change de ligne
6E1C CD624D   VARNMB:  CALL  CHINDC         Cherche si existe.
6E1F 3804              JR    C,S2
6E21 3E30              LD    A,30H          Sinon, écrit 0.
6E23 18F1              JR    S3
6E25 23       S2:      INC   HL             HL pointe sur début de la valeur.
6E26 11A172            LD    DE,#FAC1       La transporte à FACI (Floating Accumulator)
6E29 D5                PUSH  DE
6E2A 010500            LD    BC,0005H
6E2D EDB0              LDIR
6E2F E1                POP   HL             HL=FACI
6E30 CDA335            CALL  NUMPRT         Le décode et l'écrit en ascii
6E33 CDE217            CALL  ECRMES         L'écrit
6E36 18E1              JR    S4             Change de ligne.
6E38 CD0900   CTRL0:   CALL  NL             Passe à la ligne
6E3B 2AA173            LD    HL,(DEBPRO)    HL pointe sur le début du programme
6E3E 5E       BOU:     LD    E,(HL)
6E3F 23                INC   HL
6E40 56                LD    D,(HL)         DE=longueur de la ligne de programme.
6E41 7A                LD    A,D
6E42 B3                OR    E              Teste si fin du programme.
6E43 C8                RET   Z              Si oui, sortie.
6E44 EB                EX    DE,HL
6E45 19                ADD   HL,DE
6E46 2B                DEC   HL             HL pointe le début de la ligne suivante.
6E47 E5                PUSH  HL             Le sauve en pile
6E48 EB                EX    DE,HL
6E49 23                INC   HL
6E4A 4E                LD    C,(HL)         BC=N° de ligne
6E4B 23                INC   HL
6E4C 46                LD    B,(HL)
6E4D 23                INC   HL
6E4E 7E                LD    A,(HL)         HL pointe sur le début du contenu de la
6E4F FEC7              CP    C7H            ligne. Regarde si code = DEF
6E51 2803              JR    Z,+5
6E53 E1       BOU2:    POP   HL             Si oui, saut, sinon étudie la ligne
6E54 18E8              JR    BOU            suivante.
6E56 CD414F            CALL  SPAC0          Saute les espaces
6E59 FEB9              CP    B9H            Teste si PROC.
6E5B 280A              JR    Z,BOU1
```



```
6E5D FEFF                    CP     FFH
6E5F 20F2                    JR     NZ,BOU2
6E61 23                      INC    HL
6E62 7E                      LD     A,(HL)
6E63 FEC7                    CP     C7H
6E65 20EC                    JR     NZ,BOU2
6E67 E5              BOU1:   PUSH   HL
6E68 F5                      PUSH   AF
6E69 69                      LD     L,C
6E6A 60                      LD     H,B
6E6B CDDB21                  CALL   CHLASC
6E6E CDE217                  CALL   ECRMES
6E71 F1                      POP    AF
6E72 11A26E                  LD     DE,MESSP
6E75 FEB9                    CP     B9H
6E77 2803                    JR     Z,+5
6E79 11A86E                  LD     DE,MESSF
6E7C CDE217                  CALL   ECRMES
6E7F E1                      POP    HL
6E80 CD414F                  CALL   SPAC0
6E83 FE22                    CP     22H
6E85 20CC                    JR     NZ,BOU2
6E87 CDD317          BOU3:   CALL   WRIT
6E8A 23                      INC    HL
6E8B 7E                      LD     A,(HL)
6E8C FE22                    CP     22H
6E8E 2005                    JR     NZ,+7
6E90 CDD317                  CALL   WRIT
6E93 1803                    JR     +5
6E95 B7                      OR     A
6E96 20EF                    JR     NZ,BOU3
6E98 CDF917                  CALL   SAUTLN
6E9B CD896D                  CALL   ATTEND
6E9E 20B3                    JR     NZ,BOU2
6EA0 E1                      POP    HL
6EA1 C9                      RET
6EA2 2050524F        MESSP:  DEFM   " PROC"
6EA6 43
6EA7 00                      DEFB   00H
6EA8 20464E          MESSF:  DEFM   " FN"
6EAB 00                      DEFB   00H
6EAC                         END
```

Teste si FN (de code FF C7)

Pointeur sur code de FN ou PROC
A=B9 ou PROC et C7 si FN
HL=N° de ligne
Le convertit en décimal ASCII
L'écrit à l'écran
Reprend le code

Ecrit PROC ou FN selon le cas.

Revient dans la ligne de programme.
Saute les espaces.
Si on ne trouve pas «, on change de ligne

Ecrit les caractères du nom FN / PROC
jusqu'à trouver « ou la fin de la ligne
de code OOH

Change de ligne sur écran
Teste l'enfoncement de Space ou Shift Br.
Sinon recommence
Si oui, fini.

LES POSSIBILITES INSOUPCONNEES DE LA ROM DU LECTEUR DE DISQUETTES

Heureux posseseurs d'un lecteur de disquettes, il se peut que votre lecteur s'il appartient à la seconde génération, soit équipé de la dernière ROM permettant l'écriture et la lecture de chaque secteur de vos disquettes. Comme le manuel ne parle pas de ces possibilités, je vous fais part ici de quelques unes de mes découvertes.

Votre lecteur (muni de sa Master Disquette), et votre MZ-700 venant juste d'être allumés, entrez les commandes suivantes :
* JF004 et CR si le message BOOT DRIVE apparaît, répondez par la lettre D ; s'affiche alors : « sds MINIDOS on Sharp MZ-700 V2.0»

Si tel n'est pas le cas, votre lecteur n'est pas équipé de la dernière ROM, repassez par la case départ et sautez directement au paragraphe « ANCIENNES ROM »

Pour mieux comprendre la suite, sachez qu'une disquette comporte 1120 secteurs (460 en HEXA) et que les 48 premiers (30 en HEXA) sont réservés à la directory de la disquette.

1. LES PREMIERES COMMANDES

>D1,0010 et CR : affiche le secteur numéro 16 (10 hexa) dans lequel vous reconnaîtrez les noms des premiers programmes de votre disquettes. Notez qu'en appuyant sur une touche le « scrollnig » s'arrête et CR reprend ligne à ligne.

>R1, 0010, 0100, 1200 et CR : c'est la commande qui permet la lecture du secteur 10 hexa sur une longueur de 0100 hexa octets écrits à partir de l'adresse RAM 1200 hexa. Pour visualiser le résultat, faire maintenant : >M et CR : c'est le retour au moniteur de base.

Faites maintenant * D1200, vous retrouvez les informations précédentes.

2. PREMIERE APPLICATION :

Créer une disquette master auto reproductible pour cela :

• Faire une copie de la disquette d'origine, celle livrée avec le lecteur, en utilisant le programme « UTILITY »

• Reprendre les commandes « D » et « R » ci-dessus, (en respectant leur syntaxe).

• Sous le moniteur, faire D1200, s'affiche alors l'écran suivant :

```
1200 80 00 00 00 00 00 00 00  }.......
1208 00 00 00 00 00 00 00 00  ........
1210 00 00 00 00 00 00 00 00  ........
1218 00 00 00 00 00 00 00 00  ........
1220 01 55 96 A6 B8 A6 96 BD  .Utility
1228 0D 00 00 00 00 00 00 00  ........
1230 00 00 01 00 00 09 00 12  ........
1238 00 12 00 00 00 00 40 01  ......@.
1240 01 46 A6 B8 A6 B0 97 5F  .Filing←
1248 43 4D 54 0D 00 00 00 00  CMT.....
1250 00 00 01 00 00 16 00 12  ........
1258 00 12 00 00 00 00 49 01  ......I.
1260 00 4D 41 53 54 45 52 43  .MASTERC
1268 4F 50 59 0D 00 00 00 00  OPY.....
1270 00 00 00 00 00 09 00 12  ........
1278 00 12 00 00 00 00 5F 01  ......↙.
1280 00 00 00 00 00 00 00 00  ........
1288 00 00 00 00 00 00 00 00  ........
1290 00 00 00 00 00 00 00 00  ........
1298 00 00 00 00 00 00 00 00  ........
```

• Pour récupérer l'utilitaire « MASTERCOPY » (effacé d'origine) faire M1260 et répondre 01, touche CR et shift Break.

• Reprendre le Mini-Dos par JF57E et CR puis la commande d'écriture (Write) suivante :
>W1, 0010, 1200

Enfin >B (chargement du Basic par BOOT).
La commande DIR fournit « MASTERCOPY », précieux programme permettant de dupliquer n'importe quelle disquette.

REMARQUE :
Si vous rechargez un autre programme sur cette disquette, « Mastercopy » va être écrasé. Pour éviter cet inconvénient lancer les opérations suivantes :

• RUN « MASTERCOPY »
• Passer sous moniteur (par M)
• Faire S12001AFF1200 sauvegarde sur cassette de « MASTERCOPY »
• Taper # qui ramène au basic et faire :
  • UNLOCK puis DELETE « UTILITY » (désormais inutile !).
  • DELETE « MASTERCOPY »
  • RUN « FILING-CMT » et récupérer « MASTERCOPY » qui se trouve sur la cassette réenroulée.

Vous disposez maintenant d'une disquette auto-reproductible que vous pouvez compléter avec vos programmes et utilitaires favoris et dont la copie sera elle-même auto-reproductible par « MASTERCOPY ».

3. DEUXIEME APPLICATION :

Augmenter la capacité des disquettes.
Peut-être avez-vous pu constater sur une disquette Master l'impossibilité de sauvegarder un programme suite au message d'erreur 53 (plus de place sur disquette) et ce, malgré le fait que DIR vous indique au moins 136 secteurs libres.
En fait la disquette d'origine vous prive de 136 secteurs et je vous propose de les récupérer sur la disquette que nous venons de fabriquer.

• Sous le moniteur de base, faire JF57E puis :
>R1, 000F, 0100, 1200 (lecture du 15ème secteur écrit à l'adresse 1200 hexa) et CR puis >M et CR

• D1200 donne les informations suivantes :

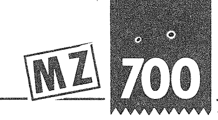
```
1200 00 B8 D7 00 60 04 FF FF  .I‾.  .II
1208 FF FF FF FF FF FF FF FF  IIIIIIII
1210 FF FF FF FF FF FF FF FF  IIIIIIII
1218 FF FF 7F 00 00 00 00 00  II  ....
1220 00 00 00 00 00 00 00 00  ........
1228 00 00 00 00 00 00 00 00  ........
```

• Faire M 1206

1206 FF 00
1208 FF 00
.
.
1217 FF 00

Mise à 00 de l'adresse 1206 à l'adresse 1217 inclues.

1218 FF shift Break



• Revenir au Mini-Dos par JF57E et écrire la modification sur disquette par : >W1, 000F, 0100, 1200 et CR

• Revenir au moniteur par >M et CR

Nous avons ainsi libéré 136 secteurs bloqués dans la directory.

ANCIENNES ROM :

Pour ceux d'entre vous qui ne possèdent pas le MINI-DOS SHARP, je vous propose de constituer, malgré cela, un utilitaire « Mastercopy » permettant de dupliquer n'importe quelle disquette du MZ-700.

Pour cela :

1. Charger par RUN le programme « UTILITY »

2. Sur l'option demandée, répondre M pour retour au Moniteur.

3. Modifier les 3 octets suivants par la commande M14D7 répondre successivement :

14D7 3A C3 + CR.
14D8 00 DE + CR.
145D9 2E 13 + CR.
14DA B7 shift + Break

4. Pour réactiver ce programme modifié faire : J1200 + CR. Vous disposez ainsi d'un programme équivalent à MASTERCOPYque vous pouvez recharger sur cassette par la commande S1200 1AFF 1200 sous Moniteur, puis sur disquette par « Filing CMT ».

REMARQUE GENERALE :

Cet exposé serait incomplet en oubliant de vous encourager à dupliquer vos disquettes avant toute investigation plus poussée (surtout de la commande Write) des possibilités du Mini-Dos.

X. PIERSON



CLAVIER AZERTY

On s'habitue fort bien au clavier QWERTY. L'inconvénient, c'est que cette habitude persiste quand on passe à un clavier AZERTY. Puisque ce dernier est usuel en France, pourquoi ne pas en doter le MZ 700 ?

Il suffit de permuter certaines données de clavier stockées entre les adresses $ 121F et $ 129C, puis de sauvegarder sur une cassette vierge le Basic ainsi modifié et, enfin, d'étiqueter les touches du clavier en fonction de leur nouvelle affectation.

Le programme ci-joint, « CLAVIER AZERTY », modifie le Basic, puis le sauvegarde. La procédure de sauvegarde est tirée du bulletin n° 9.

Le clavier résultant respecte strictement la position des lettres du clavier type AZERTY. Quant aux signes, ils ont été placés de façon que les plus couramment utilisés soient obtenus sans actionner la touche SHIFT (,:=+?()"$ -;)

On peut utiliser le programme simplement pour modifier le clavier au coup par coup, sans sauvegarder le Basic résultant. Il suffit de placer une ligne 251 END.

Il est également possible, si l'on n'utilise que l'AZERTY de permuter les touches du clavier en déboitant (délicatement) les capuchons des dites touches.

Bien entendu, la modification est sans incidence sur le Monitor ROM qui conserve les données de clavier QWERTY.

La ligne 20 contient une modification de la couleur courante (noir sur bleu pâle, qui est moins fatiguant que blanc sur bleu). On peut choisir toute autre combinaison. Le bulletin n° 8 indique que les données de couleur sont contenues dans un octet où les bits 1,2 et 3 expriment la couleur du fond et les bits 5,6 et 7 celle des caractères. Puisque au bit 5 correspnd la puissance 4 de 2, qui est aussi la puissance 1 de 16, pour exprimer les données couleur en hexadécimal il suffit de prendre les valeurs décimales des deux couleurs choisies. Ainsi, blanc (7) sur bleu (1) donne $ 71, rouge (2) sur jaune (6) donne $ 26, noir (0) sur bleu pâle (5) donne $ 05, etc.

Arnaud Samuel

```
1 '
2 'Programme "CLAVIER AZERTY"
3 '
4 'S. ARNAUD - Mars 1985
5 '
10 CLS
20 POKE $5D,$5 :'couleur noir/bleu pale
30 '
40 'Permutation des donnees de clavier
50 '
60 DATA $24,$19,$0,$4D,$2B,$D,$59
70 DATA $57,$2D,$5B,$5D,$0,$0,$0
80 DATA $41,$52,$53,$54,$55,$56,$5A
90 DATA $58,$49,$4A,$4B,$4C,$2C,$4E
100 DATA $4F,$50,$51,$42,$43,$44,$45
110 DATA $46,$47,$48,$31,$32,$33,$34
120 DATA $35,$36,$37,$38,$29,$28,$22
130 DATA $20,$30,$39,$3B,$3A,$18,$10
140 DATA $12,$11,$13,$14,$3F,$3D,$90
150 DATA $17,$FB,$5,$0,$B3,$2A,$D
160 DATA $BD,$A3,$93,$BE,$80,$0,$0
170 DATA $0,$A1,$9D,$A4,$96,$A5,$AB
180 DATA $A2,$9B,$A6,$AF,$A9,$B8,$2F
190 DATA $B0,$B7,$9E,$A0,$9A,$9F,$9C
200 DATA $92,$AA,$97,$98,$21,$94,$23
210 DATA $5C,$25,$26,$27,$FC,$C0,$C6
220 DATA $5F,$20,$FF,$5E,$2E,$3C,$16
230 DATA $15,$12,$11,$13,$14,$40,$3E
240 FOR Q=$121F TO $129C
250 READ A : POKE Q,A : NEXT Q
260 '
270 'Routine de transfert du Basic
280 '
290 DATA $AF,$11,$00,$00,$CD,$33,$00,$2A
300 DATA $71,$11,$22,$54,$12,$11,$00,$00
310 DATA $21,$79,$6B,$B7,$ED,$52,$4D,$44
320 DATA $21,$00,$12,$D3,$E0,$ED,$B0,$C3
330 DATA $00,$00,$00,$00,$00,$00,$00,$00
340 FOR Q=$6B79 TO $6BA0
350 READ A : POKE Q,A : NEXT Q
360 '
370 'Routine de transfert du Basic
380 'a sa position initiale
390 '
400 DATA $21,$22,$18,$22,$E7,$00,$3E,$00
410 DATA $32,$CF,$6B,$32,$D1,$6B,$21,$A0
420 DATA $6B,$11,$A0,$7D,$01,$A1,$6B,$ED
430 DATA $B8,$C3,$6A,$4C,$00,$00,$00,$00
440 FOR Q=$C000 TO $C01F
450 READ A : POKE Q,A : NEXT Q
460 '
470 'Renvoi au Monitor ROM et sauvegarde
480 CLS : CURSOR 5,8
490 PRINT"Faire BYE puis *GC000"
500 CURSOR 5,10
510 PRINT"Placer une cassette vierge"
520 CURSOR 5,12
530 PRINT"Faire *S12007DA07D79"
540 CURSOR 5,14
550 PRINT"(noter car l'ecran va s'effacer)"
560 CURSOR 0,23
```

CONNECTIONS PERITEL ET MONITEUR

Votre MZ-700 ou 800 est bien souvent connecté, via son cordon PERITEL, au téléviseur familial, muni en général d'un écran d'au moins 56 cm. Or, un travail prolongé à 1,20 m de cet écran se révèle rapidement fatigant et même dangereux pour les yeux. Pour palier à cet inconvénient, nous vous proposons 2 solutions :

1. FABRIQUER UN CABLE PERITEL PLUS LONG, à l'aide des 3 schémas ci-dessous, d'un fer à souder et d'un peu de patience, vous pourrez dessouder les fils de vos prises PERITEL (côté téléviseur) et DIN (côté MZ), vous munir d'un câble blindé, 8 conducteurs d'environ 3 mètres et ressouder exactement aux mêmes endroits, les fils de votre nouveau câble. N'oubliez pas de relier le corps de la fiche Din (masse), à la broche E, et de relier ce même fil de masse aux broches 2, 18, 16, 12 et 14 de la prise PERITEL.

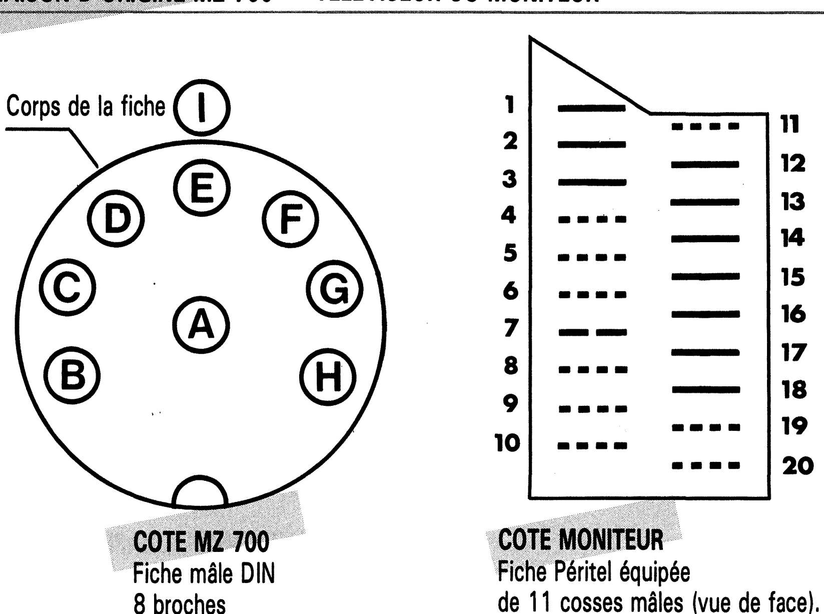
Attention, aucune des soudures, dans les 2 prises, une doit toucher sa voisine sous peine de phénomènes bizarres sur l'écran du téléviseur.

2. CONNECTER VOTRE MZ SUR UN MONITEUR VIDEO NOIR ET BLANC (ou vert ou jaune, les goûts et les couleurs...)

Là c'est beaucoup plus simple : procurez vous un câble blindé vidéo 2 conducteurs. En général, ces câbles sont vendus avec deux prises « CHINCH » à leurs extrémités. Changez l'une de ces deux prises par une fiche DIN 8 broches mâle et connectez la borne C au conducteur central et la borne E au fil de masse qui entoure généralement le conducteur central. Cette solution, bien que plus onéreuse (un moniteur coûte environ 1 000 F) vous permettra de travailler de longues heures sur votre MZ sans aucune fatigue pour vos yeux.

P. COSMIDES

SCHEMA DE CABLAGE DU CORDON DE LIAISON D'ORIGINE MZ 700 – TELEVISEUR OU MONITEUR



COTE MZ 700
Fiche mâle DIN
8 broches
(Vue de face)

COTE MONITEUR
Fiche Péritel équipée
de 11 cosses mâles (vue de face).
▬▬ : Cosse Mâle
- - - - : emplacement vide

CORRESPONDANCE DES BROCHES DE LA FICHE DIN AVEC LES COSSES DE LA FICHE PERITEL

A	⟶	17
B	⟶	15
C	⟶	3
D	⟶	-
E	⟶	2-18-16-12-14-I
F	⟶	7
G	⟶	1
H	⟶	13
I	⟶	voir (E)

ASTUCES ASTUCES



UN PETIT POKE

Un petit POKE pour transformer la première ligne de votre programme BASIC en ligne n° 0. Son intérêt c'est que vous ne pourrez plus l'effacer autrement que par DELETE.
POKE $ 6BD1, $0

F. UBALDI



ERRATA ORDRE COMPACT.

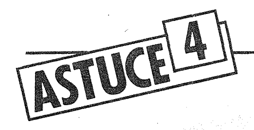
2 petits « BUGS » ont suffi à rendre l'ordre COMPACT (N° 11 page 37) totalement inopérant. Réparons vite cette erreur :
9e ligne – 6ème code – lire EB et non ED.
11 e ligne – 4 ème code – lire 18 et non 19.



SOUS DISK BASIC.

Sous BASIC disquettes, l'accès au deuxième générateur de caractères est considérablement simplifié par l'emploi d'une astuce de l'instruction COLOR.
La commande COLOR, X,0 ou X est compris entre 8 et 15 permet d'utiliser tous les caractères spéciaux directement au clavier, dans la couleur spécifiée par X-8.
Essayez par exemple :
COLOR, 15, 0 : ? « ESSAI » : COLOR, 7,0
Attention, cette astuce ne fonctionne que sous DISK BASIC.

C. SZCZEPANSKI



A PROPOS DE THEN

Nous avons vu dans le numéro précédent que THEN pouvait être omis sauf devant certains mots clés tels que « MUSIC » ou « CURSOR ». Et bien vous pouvez remplacer alors « THEN » par « : »
exemple :
IF A >B THEN MUSIC « A » = IF A >B : MUSIC « A »

G. PIDOUX



CURSEUR EN MODE GRAPH

En mode « GRAPH » les touches de curseur n'ont plus d'action sur le curseur mais affichent en vidéo inverse une flèche. Pour récupérer leur fonction initiale il suffit d'utiliser simultanément la touche « CTRL »

G. P.



TEMPS DE REACTION ET VITESSE DE REPETITION

Les touches à répétition dépendent de 2 temporisations. L'une sert à déterminer le temps de réaction (T), l'autre détermine la vitesse de répétition. Pour modifier ces paramètres il suffit de POKER comme suit :
Temps de réaction : POKE $2BC, T Valeur normale de T = $60
Vitesse de répétition : POKE $288, V Valeur normale de V = $10
T et V sont compris entre 0 et $FF toutefois il vaut mieux ne pas donner à T et V de valeur trop petite car le clavier serait incontrôlable.

G. P.

LE MZ AU DIAPASON

Mes affrontements avec l'instruction Music pour faire de mon clavier un piano m'ont rapidement convaincu que cette monture n'obéirait jamais aux caresses de mes doigts, mais serait docile uniquement sous les coups d'étriers des longueurs de notes et les coups de cravache du Tempo.

J'ai donc enfourché USR (68) et j'ai trouvé un destrier docile pour peu qu'on lui mette un nombre sous les yeux. Instantanément il bondit à la hauteur correspondant au quotient de 895000 par ce nombre et s'y maintient aussi longtemps qu'on laisse ce nombre sous son regard, mais jamais plus longtemps. Ainsi le nombre zéro le projette dans une zone de silence profond et le nombre 2034 le fait bondir exactement là où résonne le LA du diapason. Voici le schéma de génération d'un son par le MZ 700 (les explications ci-dessous n'emploient que des nombres décimaux)).

Le MZ 700 émet un signal original de 895000 Hz mais un compteur répartit cette fréquence de façon à n'envoyer au générateur de son qu'une fréquence réduite correspondant à la hauteur du son qu'on lui commande d'émettre. A cet effet il utilise un ratio que j'appellerai « Données de Répartition de Fréquence », en abrégé DRF, pour faciliter l'exposé, de sorte que la fréquence du son émis est 895000/DRF. Ainsi la DRF 2034 génère une fréquence de 895000/2034 = 532 Hz, qui est celle du MI de l'octave supérieure de notre MZ 700.

Quand on fait appel à l'instruction Music, le MZ 700 consulte à l'adresse 2449 une table contenant les DRF des notes de son octave de référence. Ainsi Music « +E » émettra le MI supérieur et si, immédiatement après, on fait PRINT PEEK (2617) et PRINT PEEK (2618), l'écran affichera 145 et 6.

Le sous-programme USR (68), lui, ne consulte pas de table mais il prend ce qu'il trouve aux adresses 2617 2618. Il faut donc s'arranger pour y placer la DRF adéquate au bon moment. Mais comme on peut y placer n'importe quel nombre compris entre 0 et 65535, les fréquences des sons émis par USR (68) iront de l'infini à 13.6 Hz, c'est-à-dire depuis les ultra-sons jusqu'à la frange supérieure des infra-sons.

Pour constituer un piano, le problème va donc consister à faire en sorte que le mouvement des doigts sur le clavier provoque en 2617 2618 l'apparition instantanée de la DRF convenable, c'est-à-dire zéro quand on cesse d'appuyer sur une touche et la DRF de la note attribué à une touche. Or la réponse à l'instruction GET place à l'adresse 95 le code ASCII du caractère généré par une touche ou bien zéro si aucune touche n'est enfoncée. Pour constater, il suffit de faire tourner le programme ci-après tout en pianotant sur le clavier :

10 GET A $: PRINT PEEK (95) : GOTO 10

Il va donc suffire de dresser à l'intention de USR (68) une table de 256 cases dnt les numéros seront les codes ASCII des caractères présents sur les touches du clavier et qui contiendront les DRF des notes qu'on aura affectées aux touches retenues. Ainsi si on décide que la touche V doit engendrer le MI supérieur, la case 86 devra contenir 1681. La case 0 contiendra obligatoirement la DRF zéro qui est celle du silence.

Il ne sera pas nécessaire de calculer toutes les DRF de tous les octaves qu'on veut rendre le clavier capable de couvrir. En effet, la fréquence d'une note d'un octave quelconque est égale à celle de la note homologue d'une octave de référence multipliée (ou divisée) par 2, 4, 8... suivant qu'on est monté (ou descendu) de 1, 2, 3... octaves. Quelques lignes de programme suffiront pour dresser la table complète des DRF. On en profitera pour lui faire scinder les DRF en deux parties telles qu'elles doiven aller respectivement en 2617 et 2618. Notre table sera donc un tableau T dimensionné DIM T (255,2). Le programme de pianotage se réduira alors à deux lignes :

100 GET N$: POKE 2617, T(PEEK(95), 2)
110 USR (68) : GOTO 100

Le programme ci-contre « PIANO » propose une procédure pour composer un clavier et pour l'utiliser. Ce ne sont pas les DRF d'une octave de référence qui y sont fournies par les DATA, mais les fréquences elles-mêmes de cette octave car les fréquences sont des points de repère universels. De plus ce sont les fréquences naturelles au sein desquelles la fréquence du LA du diapason est 440 Hz, alors que le MZ 700 utilise une gamme transposée. A partir des fréquences du programme, qui sont celles de l'octave numérotée 3 en notation française, il est possible de composer à la convenance de chacun un clavier couvrant depuis l'octave 0 jusqu'à l'octave n, en affectant les majuscules, les minuscules et les signes, mais pas les GRAPH.

Le programme est conçu pour faire des essais de composition du clavier. Quand on aura choisi la composition la mieux adaptée à ses références, il sera judicieux de transformer la partie n° 2 du programme pour fixer définitivement les affectations, de façon à éviter d'avoir à affecter les touches à chaque chargement.

S. ARNAUD

```
10 'Programme "PIANO"
20 '
30 'S. ARNAUD - Mars 1985
40 '
50 ' 1° - Explications eventuelles ****
60 '
70 CLS : CURSOR 8,12
80 PRINT"Explications (O/N) ?";
90 GET R$ :IF(R$<>"O")*(R$<>"N")THEN 90
100 IF R$="O" CLS :GOSUB 530
110 '
120 ' 2° - Composition du clavier *****
130 '
140 CLS :DIM T(255,2)
150 DATA DO,262,DO#,278,RE,294,RE#,311
160 DATA MI,330,FA,350,FA#,370,SOL,392
170 DATA SOL#,415,LA,440,LA#,466,SI,494
180 CONSOLE 12,1 :CLS
190 INPUT"Octave ? "; OCT
200 CLS :CONSOLE
210 MULT=.25*(2^(OCT-1))
220 RESTORE 150 : FOR Q=1 TO 12
230 READ NOTE$,FREQ :CURSOR 5,12
240 PRINT"J'affecte ";NOTE$;" a ";
250 GET CAR$ :IF CAR$="" GOTO 250
260 IF CAR$=" " THEN 400
270 PRINT CAR$ :FOR F=1 TO 400 :NEXT F
280 CLS :CURSOR 8,2
290 PRINT"Octave";OCT;" - "; NOTE$;
    " est affecte a ";CAR$
300 FREQ=FREQ*MULT
310 DRF=895000/FREQ
320 ND=INT(DRF/256)
330 NG=DRF-ND*256
340 T(PEEK(95),1)=NG
350 T(PEEK(95),2)=ND
360 NEXT Q :CURSOR 5,12
370 PRINT"Autre octave (O/N) ?";
380 GET R$ :IF(R$<>"O")*(R$<>"N")THEN  380
390 IF R$="O" THEN 180
400 CLS :CURSOR 8,12
410 PRINT"Le clavier est pret"
420 FOR F=1 TO 1000 :NEXT F :CLS
430 '
440 '3° -  Execution de la musique ****
450 '
460 GET N$
470 POKE 2617,T(PEEK(95),1)
480 POKE 2618,T(PEEK(95),2)
490 USR(68) : GOTO 460
500 '
510 ' 4° - Explications ***************
520 '
530 CONSOLE 0,24,5,34 :PRINT"";
540 PRINT"Pour composer le clavier :"
550 PRINT"Choisissez sur le clavier la"
560 PRINT"position la plus favorable a"
570 PRINT"l'octave de reference (octave 3)"
580 PRINT"Commencez par cette octave"
590 PRINT"en repondant 3"
600 PRINT"Pour affecter les touches,"
610 PRINT"enfoncez-les sans faire CR"
620 PRINT"Pour arreter en cours d'octave"
630 PRINT"appuyez sur la barre"
640 PRINT"d'espacement"
650 PRINT"Enfoncez une touche";
660 GET X$ :IF X$="" GOTO 660
670 CONSOLE : RETURN
```

SUPER BASIC 6

Le K. BASIC est maintenant reconnu comme étant le BASIC le plus puissant implenté sur un ordinateur « familial » il est utilisé et unanimement apprécié par la grande majorité des possesseurs de MZ 700 tant amateurs que professionnels. La version MZ 800 conserve les mêmes structures de base que le K. BASIC. Ont été ajoutées : la gestion des QUICK DISC, la haute résolution graphique et la possibilité de redéfinir tous les caractères affichables.

Les principales commandes spécifiques ont été décrites, lors de la présentation du K. BASIC, dans notre bulletin numéro 10. Ce BASIC est dès maintenant disponible sur QUICK DISC, accompagné de nombreux programmes permettant de tirer un meilleur profit de ses immenses ressources et d'une notice très détaillée dotée de nombreux exemples.

Vous pouvez vous le procurer au Club au prix de 180 F. (Quick Disc, notice et port compris)

Parallèlement à la conception de ce nouveau langage, Monsieur KOKANOSKI est également l'auteur, dans ce bulletin, de 3 articles dont la qualité n'est plus à démontrer : un langage PILOT pour MZ 80K, une tortue graphique pour MZ 700 et de nouvelles améliorations à son K. Basic.

L'apport considérable des compétences de Monsieur KOKANOSKI a contribué, pour une large part à l'essor et à l'évolution de notre Club, pour le plus grand bien de tous. Qu'à cette occasion, il nous permette de lui renouveler nos très sincères remerciements.

LISTE DES COMMANDES AJOUTEES AU BASIC SHARP

DISP, ENDPROC, ENDCASE, ELSIF, ENDIF, DOKE, DO, INKEY, HSET, KEY, MERGE, OF, PROC, REPEAT, UNTIL, WHILE, WEND, DEF, RESULT=, BEEP, SEARCH, MOVE, DRAW, EXC, PWAIT, LOCAL, CASE, WHEN, OTHERWISE, EXIT, SPOKE, VARTPR, MAX, MIN, DIV, MODULO, COMPACT, FRAC, ASN, ACS, SH, TH, ASH, ACH, ATH, FACT, DEEK, INV$, LCURSX, LCURSY, HCURSX, HCURSY, EVAL, TRUE, FALSE, IN$, INSTR, STRING$.

L'interpréteur Super-Basic 6 est destiné à la programmation du Sharp MZ-800 muni de l'unité de lecture des Quick-Discs, uniquement en mode 700. Ce Basic est compatible avec le K-Basic V.5 puisqu'il en est une extension et une adaptation. Il sera donc possible d'utiliser tous les programmes écrits en K-Basic sur MZ-700, après les avoir transférés sur Quick-Disc.

Le Super Basic 6 se présente sous deux aspects :

- C'est tout d'abord un interpréteur Basic « standard » complet, possédant les instructions ELSE pour les tests, INKEY, KEY, GET pour « attraper » des caractères au clavier, MAX, MIN, OR, AND, XOR, NOT, DIV, MODULO,..., pour manipuler des nombres ou des chaînes de caractères, MOVE, DRAW, POINT pour le semi-graphisme ou le graphisme sur l'écran, SWAP, NULL, COPY pour manipuler les tableaux (à moins de 128 dimensions cependant !) etc.
- C'est surtout un interpréteur destinée à écrire des programmes selon les règles de la « Programmation structurée ». Dans ce type de programmation, bien connue des utilisateurs de langages de haut niveau, comme PASCAL, le programme est construit de façon modulaire, chaque module pouvant être mis au point séparément.

Avec un peu d'habitude, on pourra écrire de longs programmes, faciles à mettre au point, à modifier et dont la structure (et par la suite, le mode de fonctionnement) se reconnaît au premier coup d'œil sur le listing. On ne trouvera surtout plus ces nombreux GOTO renvoyant d'un bout à l'autre du programme et le rendant incompréhensible à tout autre qu'à l'auteur et même à celui-ci, quelques mois plus tard !

Rassurons tout de suite l'utilisateur n'ayant jamais programmé de façon structurée : le Super Basic 6 permettant les deux types de programmation, il lui sera toujours possible d'écrire son programme en suivant les mauvaises habitudes dûes à l'utilisation des Basics standards, puis le modifier, l'améliorer pour lui donner une forme structurée. Ainsi, petit à petit, il lui sera possible de s'habituer à la programmation structurée et de se rendre compte de sa grande puissance.

Ce genre de programmation est permis par la présence de :

3 types de boucles :

```
FOR       NEXT
REPEAT    UNTIL
WHILE     WEND
```

2 types de tests :

IF...ELSIF...ELSE...ENDIF...CASE...OF...WHEN...OTHERWISE...ENDCASE...

(Ces deux types de structures peuvent s'étendre sur un nombre quelconque de lignes de programme)

2 types de sous-programmes (en plus des GOSUB-RETURN classiques) :

Procédures définies par DEF PROC et appelées par PROC.

Fonctions définies par DEF FN et appelées FN.

Ces structures permettent toutes deux le passage de paramètres, par valeur et par référence, avec, bien entendu, l'utilisation possible de variables locales par LOCAL. Elles permettent aussi la récursivité.

Bernard Kokanosky

UN MONITEUR PEUT EN CACHER UN AUTRE

Le moniteur du MZ-800 possède un certain nombre de commandes non décrites dans le manuel. Nous allons les passer en revue et indiquer comment utiliser certains sous-programmes pour manipuler directement, en langage machine, cassettes et Quick-Discs.

Le moniteur MZ-800 est constitué de deux parties, une partie « basse » allant des adresses 0000 à IIFF et d'une partie haute de EOIO à F386. La partie basse est identique au moniteur du MZ-700 dont le listing était donné dans le manuel MZ-700 à part les trois octets qui au lieu d'être C3 4A 00 (JP 004A) sont C3 00 E8 (JP E800). Le point d'entrée du moniteur haut est donc E800.

Il s'agit en fait du point d'entrée à froid, c'est à dire qu'il y a initialisation de l'ordinateur avec test de la présence d'une disquette dans le lecteur ou d'un Quick-Disc et lancement (si oui) du premier programme de la disquette. Si ce n'est pas le cas, on se retrouverera au point d'entrée à chaud (EA5E) auquel on peut accèder à partir de E804 (où l'on trouve C3 5E EA ou JP EA5E).

Trois autres sous-programme utiles ont leur point d'entrée juste après :

E807 : entrée d'un sous programme lisant un programme sur cassette et le plaçant à partir de 1200H. Le programme ne sera pas éxécuté.

E809 : entrée d'un sous-programme enregistrant sur cassette le programme qui vient d'être chargé.

E80D : entrée d'un sous-programe vérifiant ce qui vient d'être enregistré.

On accède à ces sous-programme par la commande G (GE807 par exemple). Ces programmes permettent, en particulier, de faire une copie du Basic sur cassette.
On trouve ensuite à E810 : C3 IO EO (JP EOIO) programme manipulant les Quick Disc et dont nous reparlerons.

LES COMMANDES DU MONITEUR SONT :

J saut. Exemple J000 fait repartir à l'adresse 0000 donc tout se passe comme si on venait de remettre l'ordinateur en marche.

G Appel de sous-programme. Il y a donc saut à l'adresse indiquée, mais retour là où on se trouvait avant l'appel, lors du RET du programme en langage machine.
Exemple : GE807 pour charger la cassette Basic.

B Clavier sonore ou pas

L Load. Cette commande permet de charger un programme venant d'une cassette. Il est décrit dans le manuel.

S Save permet d'enregistrer un morceau de mémoire sur cassette.

V Verify permet de vérifier ce qui a été enregistré par S.

D Dump permet de lister le contenu de la mémoire. Cette commande est décrite dans le manuel.

M Modification éventuelle de la mémoire. Voir le manuel.

F Floppy permet de passer le contrôle au programme gérant les disquettes.

EB (Extension Boot ?) permet d'envoyer le contenu de la mémoire centrale dans la RAM supplémentaire.

QL Chargement d'un programme en langage machine depuis le Quick-Disc.

QS Sauvegarde sur le programme. Dans ce cas, le moniteur demande, en plus du nom, l'adresse de début (Top adrs), de fin (End adrs) et d'éxécution du programme.
Il n'y a pas de commande Verify, puisque le programme d'enregistrement vérifie lui-même que la sauvegarde a été correcte (Si elle ne l'était pas, il y aurait 5 tentatives d'écriture-vérification, avant l'apparition d'un message d'erreur !)

QF permet de formater un Quick-Disc

QD donne le contenu (Directory) du Quick-Disc

QX permet de charger un programme en mémoire depuis la cassette et de l'enregistrer sur le Quick-Disc.

QC Copy permet de recopier un programme d'un Quick-Disc sur un autre.

La gestion de la mémoire (M,D) et des cassettes (L,S,V) étant bien connue, je vais essayer d'exposer ce que j'ai trouvé au sujet de la manipulation des Quick-Discs, en désassemblant le moniteur haut.

Avant toute chose, disons qu'il existe un sous programme gérant les erreurs, c'est à dire principalement, affichant les divers messages d'erreur. Il se trouve à F202H (ERRORS). L'arrivée dans ce sous-programme se fait avec A contenant un code d'erreur :

A = 28 File not found
A = 39 Bad disk err
A = 2E Write protect
A = 32 Not ready
A = 00 Break
A = 35 No file space err
A = 36 Unformat err
A = 2A Already exist err
A = 33 Too many files err (32 au maximum).
Autres valeurs : Hard err

La sortie de ce sous-programme est intéressante car elle a visiblement été conçue pour une utilisation par l'amateur de programmes en langage machine. En effet la sortie dépend du contenu de la mémoire II3AH.

Si II3AH = 1 le retour se fait par RET sans qu'aucun message d'erreur soit écrit.
Si II3AH = 2 le retour se fait par RET mais le message est écrit.
Sinon le retour se fait au moniteur (départ à chaud) par JP EA5E.

Signalons aussi que ce sous-programme arrête le moteur du Quick-Disc.

Il est bon aussi, avant d'utiliser l'unité de Q.D, vérifier si elle est bien connectée. Le sous-programme EB13H retourne Z si l'unité est connectée et NZ sinon. enfin, avant toute opéra-

tion (chargement ou écriture), il faudra appeler le sous-programme EEECH qui vide 3 mémoires chargées d'obliger la tête de lecture/écriture à se replacer au début du Quick-Disc lors de la prochaine opération, et de garder trace de deux pointeurs utilisés pour l'écriture de fichiers en plusieurs morceaux.

Maintenant, toutes les opérations sur Quick-Disc se feront par le sous-programme (E010H ou E810H), le comportement de l'ordinateur différant selon la valeur contenue dans la mémoire 1130H. Voici ce que produit CALL E010H selon les valeurs de 1130 :

1130H = 1 L'ordinateur teste si le Quick-Disc est prêt pour une opération de lecture (si 1131 contient 0) ou d'écriture (si 1131 ≠0). Le retour se fera Non Carr si oui et Carry (avec le code d'erreur dans A) sinon.
On peut utiliser directement cette possibilité : CALL EFE1H s'il est prêt pour une écriture.

1130H = 2 Il y aura formatage du disque. On trouve cela dans le sous-programme réalisant la commande QF (débutant à F0B5H). Retour Carry avec A = 29 si impossible

1130H = 5 C'est le sous-programme (F25F). Ce sous-programme obligera, lors de la prochaine lecture, la tête de lecture à reprendre le disque au « début ». Cette opération est essentielle lors de la lecture de tout un fichier.

1130H = 6
ou > 6 Arrêt le moteur et une mémoire qui obligera la tête à reprendre le disque au « début » lors de la prochaine opération de lecture / écriture.
CALL E2E8H. A sera conservé par le sous-programme.

Nous allons voir maintenant les cas où la valeur est 4 (SAVE) ou 3(LOAD).
Pour SAVE, le sous-programme réalisant la commande QS et débutant à EF2EH nous apprend beaucoup de choses.

Tout d'abord, il teste si le QD est prêt pour une écriture. Si Carry, non, donc saut au traitement des erreurs. Si oui, il demande le nom du fichier et va le placer (seul 16 caractères au maximum sont autorisés) à partir de 11A3H. Le sous-programme est à EC8DH. Malheureusement, il ne peut pas être utilisé tel quel par l'amateur pour son propre usage puisque si l'on fait Shift Break, il y aura retour au moniteur. Il faudra donc recopier ce sous-programme et modifier les octets EC9F,ECA0, ECA1 et y mettre LD HL, adresse de retour (21 xx xx) au lieu de 21 5E EA qui fait retourner au moniteur. Le programme teste ensuite si on a entré un nom « vide » ce qui est interdit. Le nom est ensuite transféré aux octets 10F1H et suivants.

Il demande ensuite l'adresse de début (placés dans 1106H et 1107H), l'adresse de fin ce qui permet de placer la longueur dans 1104H et 1105H et enfin, l'adresse d'éxécution qui est placée à 1108H et 1109H. L'amateur averti remarquera qu'il y a un décalage de deux octets par rapport au mode de gestion des cassettes.

Le code 01H est placé dans 10F0H (programme en langage machine).

Ensuite, le programme teste s'il existe déjà un programme du même nom sur le disque. C'est le but du sous-programme EF9F. Retour Carry si existe déjà. S'il y a eu une erreur de fonctionnement, le retour se fait Non Carry avec A ≠28H d'où un saut au traitement des erreurs.

Par contre, si A = 28H, cela signifie que le programme n'existe pas. On peut donc le charger sur le disque.

On placera alors dans les mémoires :
1132 et 1133 l'adresse du « titre » du programme (ici 10F0H)
1134 et 1135 la longueur de ce titre (0040H pourquoi???)
1136 et 1137 l'adresse du début de la zone à enregistrer
1138 et 1139 la longueur à enregistrer.

En plaçant 04 dans 1130 et 04 dans 1131 et en faisant CALL E010H, le programme a enregistré en deux morceaux, le premier contenant le titre et le second, le programme proprement dit. Le retour se fera Non Carry si tout est correct et Carry s'il y a eu un problème (code d'erreur dans A).

On remarquera que l'adresse du début du programme se trouve aussi dans 1106H et 1107H, sa longueur dans 1104 et 1105 et l'adresse d'éxécution dans 1108 et 1109. Toutes ces valeurs sont enregistrées avec le titre (nom et code 01 dans 10F0H).

Cependant, rien n'empêche de donner à 1136, 1137, 1138, 1139 des valeurs différentes de celles de 1106, 1107, 1104, 1105, ce qui peut dans certains cas permettre l'enregistrement de programmes qui auraient été déplacés en mémoire.

L'examen plus complet du moniteur montre qu'il a été construit de façon à permettre l'enregistrement de longs programmes, par morceaux. Supposons, pour fixer les idées que nous voulions enregistrer sur disque, la taille totale, l'adresse d'éxécution et le nom, et après avoir vérifié que le disque ne contenait pas un fichier (ou programme) de même nom, on procèdera comme suit.

On placera dans 1132, l'adresse du titre (comme précédemment)

Puis dans 1134 et 1135, la longueur de ce titre (code, nom, taille, adresse de début,...)

On placera ensuite dans 1136, 1137 l'adresse du début du premier morceau à enregistrer dans 1138, 1139 la taille de ce morceau.

On placera 04 dans 1130H et 00 dans 1131H.

On fera CALL E010H ce qui va enregistrer sur le disque, un code 00 suivi du titre, puis un code 01 suivi du premier morceau. Pour le second morceau, on placera dans 1136, 1137, 1138, 1139 son adresse de début et sa taille.

On placera 01 dans 1131H et on fera CALL E010H, ce qui va enregistrer le second morceau, à la suite du précédent (avec un code 01 devant).

Pour le dernier morceau, on remplira 1136, 1137, 1138, 1139 comme précédemment, et on placera 05H dans 1131H. Call E010H enregistrera alors le dernier morceau précédé d'un code 05. De plus, le fichier ainsi construit sera « fermé » et considéré comme un seul bloc. S'il y a eu une erreur quelconque au cours de l'enregistrement, tout le fichier sera effacé.

En résumé, ce sont les bits 0 et 2 de l'octets 1131H qui contrôlent le type d'enregistrement :
bit 0 à 0 : il y a un titre à enregistrer
bit 0 à 1 : il n'y a pas de titre à enregistrer.
bit 2 à 0 : on enregistre un morceau mais le fichier reste prêt à recevoir d'autres données
bit 2 à 1 : on enregistre le dernier morceau. Le fichier sera ensuite fermé.

On remarquera qu'il n'est pas possible d'enregistrer seulement le titre : il y aura toujours enregistrement du titre et du premier morceau du programme.

VOYONS ENFIN, LA COMMANDE LOAD.

Commençons par décortiquer la commande QL (le programme débute à EEA7H).

On teste évidemment si le Q.D. est prêt pour une lecture. Si oui, on demande le nom du fichier et il est écrit à partir de 11A3H. Le programme (CALL F25F) est utilisé. Il obligera la tête de lecture à reprendre le disque au début.

Puis le disque est exploré : il y a chargement à partir de 10F0H des titres des divers programmes trouvés sur le disque. Voyons déjà comment sont chargés ces titres (EEF7H).

On place dans 1132, 1133 l'adresse de l'endroit où envoyer le titre (ici 10F0H) et dans 1134, 1135 la longueur de ce titre (ici 0040H). En fait ce qu'on écrit ici, doit être supérieur ou égal à ce qui est inscrit sur le disque, sinon il y aura refus de faire le chargement. On place 03 dans 1130 et 00 dans 1131H. CALL DISKMP fait le chargement.

Lorsque le bon nom a été trouvé sur le disque. En plaçant maintenant 01 dans 1131H, CALL E010H fera le chargement. Dans le cas où on doit lire un programme coupé en plusieurs morceaux, on pourra, en faisant la même dernière opération, charger le second, le troisième,..., morceau. Le moniteur garde trace de la position du dernier fichier lu.

En résumé, pour LOAD, on place 03 dans 1130, l'adresse dans 1132, 1133, la longueur de chargement dans 1134, 1135.

Pour charger un titre, on place 00 dans 1131 et CALL E010H

Pour charger des programmes : 01 dans 1131 et CALL E010H.

Pour terminer cette rapide description, précisons que lorsqu'un programme lu sur cassette, disquette ; Quick-Disc, Ram optionnelle, est lancé, l'adresse d'éxécution se trouve dans 1X et surtout C=00 et B=00 (si RAM), B=01 (si cassette), B=02 (si Floppy Disc), B=03 (si Quick-Disc)

B. Kokanoski



PICO D.A.O.

Pourquoi PICO ? Parce que ce programme doit être considéré comme une trame de base qui vous permettra de créer votre propre D.A.O. (Dessin Assisté par Ordinateur) personnalisé au gré de votre imagination. Cette version, très dépouillée possède néanmoins des instructions puissantes spécifiques au MZ 800.

Entrez le programme puis tapez RUN. L'écran apparaît avec 5 cases de fonctions à gauche, 16 cases de couleurs en bas et un curseur graphique. Il vous suffira ensuite de déplacer ce curseur sur la case correspondante et de valider pour utiliser la couleur ou la fonction choisie. La case supérieure permet de modifier la couleur du dessin en cours. Le curseur dans cette case puis dans la case de couleur choisie effectue cette modification. La case marquée d'un visage permet de dessiner directement à l'écran. L'action simultanée de la barre d'espace et des touches curseur allume ou éteint le point correspondant au curseur. La case « / » permet de tracer des droites. Le case « O » dessine des cercles. La case « BOX » trace des cadres ; et la case « PI » colorie une zone écran délimitée précédemment. Toutes ces commandes possèdent un contrôle visuel du graphisme recherché ; seul le relachement de la barre d'espace validera le dessin. Une première modification de la variable J de la ligne 60 vous permettre d'utiliser le JOYSTICK 1 (J=1) ou 2 (J=2).

Ce programme, très court, démontre qu'une utilisation judicieuse des nombreuses possibilités du BASIC MZ 800 permet de faire beaucoup en peu de lignes.

```
10 '*********************************
20 '***           D.A.O.          ***
30 '***   Christian MARTIN 06/85  ***
40 '*********************************
50 '
60 '*********************************
70 '*****     INITIALISATION    *****
80 '*********************************
90 X=32:Y=176:C=2:M=0:J=0:GA=1:BA=198:DO=318:HA=1
100 INIT"CRT:M2"
110 BOX[1]0,0,28,178:BOX[1]30,0,319,178:BOX[1]30,180,319,199:BOX[1]0,180,28,199
120 RESTORE140:FORI=25TO173STEP30.6:READD$:BOX[1]4,I,24,I+27.6:SYMBOL[1]7,I+3,D$
,2,3:NEXT:BOX[1]5,3,24,20
130 PAINT[C]6,4,1
140 DATA"↓","/","●","▩","π"
150 FORI=43TO312STEP17:BOX[1]I,185,I+15,194:NEXT
160 FORI=0TO15:PAINT[I]45+(I*17),187,1:NEXT
170 '*********************************
180 '*****   MAIN ROUTINE        *****
190 '*********************************
200 F=POINT(X,Y):SET[15]X,Y
210 A=STRIG(J):B=STICK(J)
220 IF(X<31)AND(A=1)THENGOSUB"MOD":X1=X:Y1=Y:GOTO310
230 IF(Y>177)AND(A=1)THENGOSUB"COUL":X1=X:Y1=Y:GOTO310
240 X1=X:Y1=Y
250 ONBGOSUB 360,370,380,390,400,410,420,430,250
260 IF(A=1)AND(M=0)THEN SET[C]X1,Y1:GOTO200
270 IF(A=1)AND(M=1)THENGOSUB"LIGNE"
280 IF(A=1)AND(M=2)THENGOSUB"CERCLE"
290 IF(A=1)AND(M=3)THENGOSUB"BOITE"
300 IF(A=1)AND(M=4)THENGOSUB"REMPL"
310 SET[F]X1,Y1:GOTO200
320 '*********************************
330 '*****     SOUS-PROGRAMS     *****
340 '*********************************
350 '------      DIRECTIONS  ------
360 Y=Y-1*-(Y>HA):RETURN
370 Y=Y-1*-(Y>HA):X=X+1*-(X<DO):RETURN
380 X=X+1*-(X<DO):RETURN
390 X=X+1*-(X<DO):Y=Y+1*-(Y<BA):RETURN
400 Y=Y+1*-(Y<BA):RETURN
410 Y=Y+1*-(Y<BA):X=X-1*-(X>GA):RETURN
420 X=X-1*-(X>GA):RETURN
430 Y=Y-1*-(Y>HA):X=X-1*-(X>GA):RETURN
440 '------      CHOIX MODE  ------
450 LABEL"MOD"
460 IF(X<5)OR(X>23)THENRETURN
470 IFF=0THENF=4
480 IF(Y>27)AND(Y<50)THENPAINT[0]7,M*30+28,1:PAINT[4]7,28,1:M=0:RETURN
490 IF(Y>57)AND(Y<81)THENPAINT[0]7,M*30+28,1:PAINT[4]7,58,1:M=1:RETURN
500 IF(Y>88)AND(Y<111)THENPAINT[0]7,M*30+28,1:PAINT[4]7,89,1:M=2:RETURN
510 IF(Y>117)AND(Y<142)THENPAINT[0]7,M*30+28,1:PAINT[4]7,118,1:M=3:RETURN
520 IF(Y>149)AND(Y<173)THENPAINT[0]7,M*30+28,1:PAINT[4]7,149,1:M=4:RETURN
530 RETURN
540 '------   Chmt COULEUR  ------
550 LABEL"COUL"
560 IF(X<43)OR(X>313)OR(Y<185)OR(Y>194)THENRETURN
570 C=POINT(X+1,Y):C1=0:IFPOINT(8,8)=0THENBOX[1]5,3,24,20:C1=1
580 PAINT[C]8,6,C1:RETURN
590 '------      TRACE LIGNE ------
600 LABEL"LIGNE":GA=31:BA=177
610 XF=X:YF=Y
620 A=STRIG(J):IFA=0GOTO620
630 ONSTICK(J)GOSUB 360,370,380,390,400,410,420,430,250
640 LINE[C]XF,YF,X,Y
650 IFSTICK(J)=0GOTO650:ELSEBLINEXF,YF,X,Y
660 IFSTRIG(J)=0THENLINE[C]XF,YF,X,Y:GA=1:BA=198:RETURN
670 GOTO620
680 '------      TRACE CERCLE -----
690 LABEL"CERCLE"
700 XF=X:YF=Y:DO=200
710 V1=X-31:V2=177-Y:V3=318-X:V4=Y-1:X=0:Y=0
720 IFV1<DOTHENDO=V1
730 IFV2<DOTHENDO=V2
740 IFV3<DOTHENDO=V3
750 IFV4<DOTHENDO=V4
760 GA=-DO
770 A=STRIG(J):IFA=0GOTO770
780 ONSTICK(J)GOSUB 360,370,380,390,400,410,420,430,250
790 CIRCLE[C]XF,YF,ABS(X)
800 IFSTICK(J)=0GOTO800:ELSECIRCLE[0]XF,YF,ABS(X)
810 IFSTRIG(J)=0THENCIRCLE[C]XF,YF,ABS(X):X=XF+X:Y=YF+Y:GA=1:DO=318:RETURN
820 GOTO770
830 '------      TRACE BOITE  -----
840 LABEL"BOITE"
850 XF=X:YF=Y:BA=177:GA=31
860 A=STRIG(J):IFA=0GOTO860
870 ONSTICK(J)GOSUB 360,370,380,390,400,410,420,430,250
880 BOX[C]XF,YF,X,Y
890 IFSTICK(J)=0GOTO890:ELSEBOX[0]XF,YF,X,Y
900 IFSTRIG(J)=0THENBOX[C]XF,YF,X,Y:BA=198:GA=1:RETURN
910 GOTO860
920 '------      REMPLISSAGE  -----
930 LABEL"REMPL"
940 PAINT[C]X+1,Y,1,2,3,4,5,6,7,8,9,10,11,12,13,14,15:F=C:RETURN
```

LM SUR MZ

Cette 4e leçon va nous permettre (enfin !) d'entrer dans le vif du sujet. Nous allons élaborer notre premier programme LM et le décortiquer, instruction par instruction. Ensuite, il vous faudra relever les manches car vous aurez à taper, en HEXA l'un des utilitaires indispensables à tout programmeur LM : le désassembleur Z80.

LE PROGRAMME

Comme « Compte » de notre première leçon, son inutilité flagrante n'a d'égal que son intérêt pédagogique. Il consiste à emplir l'écran du caractère de la touche tapée au clavier. La figure 1 présente son organigramme ; la figure 2 présente le fonctionnement de cet organigramme sous Basic ; la figure 3 présente le programme LM entré à l'aide d'un assembleur.

Passons à son étude.

CD 1 B 00. CALL 00 1B

CALL (codé CD), en LM, c'est exactement le GOSUB du Basic ; mais au lieu de lui indiquer un numéro de ligne, il faut lui spécifier l'adresse à laquelle démarre le sous-programme appelé. Ce sous-programme devra impérativement se terminer par l'instruction RET (C9) qui provoque un retour à l'instruction qui suit immédiatement le CALL. Tout cela n'est pas nouveau, nous connaissons parfaitement cette structure en Basic. CALL est composé d'un octet de commande (CD) et de l'adresse 16 Bits appelée, codée sur les 2 octets suivant. Bref, un rêve de simplicité et de logique... eh bien non ! car une règle aussi élémentaire qu'empoisonnante du LM veut que toute l'adresse 16 Bits codée sur 2 octets soit écrite à l'envers. Vous remarquerez en effet que les codes machine de l'instruction CALL 001B sont CD 1B 00 ; donc inversés. Ecrire CD 00 1B reviendrait à demander au Z80 d'aller lire un sous programme en 1B00 et nous conduirait irrémédiablement vers un superbe plantage dans le cas de notre programme. Cette règle doit toujours rester présente lorsqu'on programme directement en codes machines. En assembleur, le programme assembleur se charge d'inverser automatiquement les octets.

L'adresse 001B appelée représente le point de départ de l'un des multiple utilitaires contenus dans le moniteur ROM. Celui-ci a la même fonction que l'ordre GET du Basic. Au retour de ce sous-programme, le registre A contiendra le code ASCII de la touche tapée au clavier ou 0 si aucune touche n'a été tapée. Avant de continuer, il nous faut donc savoir si A contient une valeur différente de 0 ; c'est le rôle de

CP 00. FE 00

CP (codé FE) est une instruction de comparaison qui va nous indiquer si le registre A est égal à 0 ou non. L'instruction CP 00 demande donc au Z80 de comparer la valeur de A à 0. Pour effectuer cette opération, le microprocesseur va soustraire la donnée qui suit la commande CP, et indiquer le résultat de cette opération dans les bits du registre F. F, c'est le registre d'indicateurs que nous avons survolé dans une précédente leçon. La valeur de 6 et ses 8 bits (0 et 1) dépend du résultat de l'opération précédente (voir encadré). Dans notre cas, le bit Z nous intéresse plus particulièrement. Après une opération, il prend la valeur 1 si le résultat est nul ou la valeur 0 en cas de résultat différent de 0, c'est le bit Z qui conditionne l'instruction suivante :

JP Z, C000. CA 00 C0

Dans l'instruction JP Z, il faut considérer 2 fonctions différentes : la commande JP et la condition Z. JP est une instruction de saut très similaire au GOTO du Basic. Comme pour CALL, il faut lui spécifier, sur 2 octets INVERSES, l'adresse à laquelle le programme devra se poursuivre. Dans notre cas, la commande JP est suivie de Z, ce qui veut dire que le saut ne s'effectuera que si notre bit Z est à 1, ... donc si le résultat de l'opération précédente conduit à un résultat nul... donc si A est égal à 0 ... donc si aucune touche n'a été pressée au moment de l'appel du sous-programme GET. Si Z=0 (touche pressée), le Z80 ignorera cette commande de saut et passera à la commande suivante. Nous avons donc une boucle d'attente, d'un caractère dont nous sortirons dès qu'une touche aura été détectée.

Au moment de cette sortie, a contient toujours le code ASCII de la touche pressée. Les codes d'affichage à l'écran étant différents, sur MZ, des codes ASCII, nous allons devoir modifier le contenu de A en conséquence ; c'est le rôle de :

CALL OBB9. CD B9 OB

OBB9, c'est justement l'adresse d'un autre sous-programme utilitaire du moniteur ROM qui se charge de cette tâche. Lors de son appel, le registre A doit contenir une valeur ASCII, à son retour, A contiendra le code d'affichage correspondant. Notez que CALL (CD) est toujours suivi de l'adresse d'appel inversée (B9 OB pour l'appel à OBB9).

Le travail suivant va consister à mettre dans chaque adresse mémoire vidéo le contenu de A.

LD DE, D000. 11 00 D0

LD (de LOAD) est certainement l'instruction la plus utilisée en LM ; elle permet d'entrer une valeur quelconque dans tout registre 8 bits ou 16 bits ou de transférer le contenu d'un registre dans un autre. Dans l'instruction qui nous concerne, le registre DE (16 bits) est chargé par la valeur D000 qui correspond à la première adresse de la mémoire vidéo.

LD HL, D3E8. 21 E8 D3

La valeur D3E8 est placée dans le registre HL ; elle correspond à la dernière adresse vidéo. D3E8 – D000 = 03E8 (1000 en décimal), cela correspond à notre écran de 25 lignes40 caractères.

LD (DE), A. 12

Encore une instruction de transfert de données qui, à la différence des 2 précédentes fait intervenir une notion de pointeur indiquée par les () placées autour de DE. Le contenu de A sera, en effet placé dans l'adresse ayant pour numéro DE. DE contenant actuellement la valeur D000, A sera donc placé dans l'adresse D000. DE ne sera pas modifié puisqu'il ne s'agit pas d'un transfert direct, mais servira uniquement de pointeur indiquant le numéro d'adresse où doit être transféré le contenu de A. A noter également que A, comme tout registre émetteur lors d'un transfert de données, conserve sa valeur d'origine après le transfert. Pour nous résumer plus concrètement, nous venons de placer dans le coin supérieur gauche de l'écran le caractère de la touche tapée au clavier. Il ne nous reste qu'à répéter cette opération 999 fois...

INC DE. 13

Une nouvelle instruction qui ne demande qu'un seul octet et qui pourrait se traduire, en Basic

par DE = DE + 1. INC (de incrémenter) signifie donc ajouter 1 au registre considéré, ici DE dont le contenu vaut maintenant D001. L'instruction INC ou son inverse DEC (de décrémenter : soustraire 1) s'applique à tous les registres généraux du Z80. Cette instruction nous permet, dans le cas présent de progresser d'une case dans la RAM vidéo.

SBC HL, DE. ED 52

SBC, c'est une instruction de soustraction avec retenue (de Substracht with Carry) effectuée généralement entre 2 registres 8 ou 16 bits. Nous reviendrons ultérieurement sur la retenue.

La logique de ce type d'instruction est la suivante : le contenu du second registre de l'instuction (DE dans notre cas) est soustrait au contenu du premier registre (HL) et le résultat est placé dans ce premier registre. En Basic cela nous donnerait HL = HL – DE. HL est donc modifié mais pas DE (le second registre). Comme lors de toute opération mathématique, le Z80 positionne les registres de F en fonction du résultat de l'opération. Le bit Z de ce registre va encore nous être utile pour la suite du déroulement de ce programme;

JP NZ, C00A. C2 0A C0

Nous retrouvons ici une instruction très proche du JP Z, que nous avons déjà étudié. La seule différence est que la condition d'exécution de cette instruction va être inverse. Le saut sera effectué si, après test du bit Z, celui ci indique que le résultat de la soustraction précédente n'est pas égal à zéro (NZ = No Zéro), donc si ce bit est à zéro. La soustraction précédente et ce test permettent de savoir si DE, par incréments successifs, est arrivé à la valeur D3E8. Si ce n'est pas le cas, HL-DE donnera un résultat positif et provoquera une nouvelle boucle d'écriture d'une adresse en RAM. Si c'est le cas, la condition NZ ne sera pas respectée, la commande de saut ne sera pas exécutée èt le microprocesseur passera à l'instruction suivante...

JP C000. C3 00 C0

Si l'on a bien compris le mécanisme des instructions précédentes, il est facile d'en conclure que le programme n'aboutira à cette instruction que lorsque les 1000 adresses de la RAM vidéo auront reçu le contenu de A. Nous avons étudié les sauts conditionnels, nous allons terminer par le plus facile : Ce JP saute en effet, sans « réfléchir » à l'adresse qui lui est indiquée. C'est exactement le GOTO du Basic. L'adresse C000 correspond au début du programme ; nous repartons donc pour un nouveau cycle d'attente de caractère et d'affichage et ce, jusqu'à épuisement de l'opérateur et interruption par un fatal ALL RESET. Fatal n'est pas le mot puisque, contrairement à une idée reçue, un RESET n'efface pas les programmes en mémoire. Pour vous en convaincre, il vous suffira de retaper, sous moniteur : JC000 pour relancer ce programme ; ou DC000 pour en lister les codes HEXA.

Pour clore ce chapitre et répondre à une question très souvent posée au Club, nous allons passer en revue tous les moyens d'entrer ce programme LM dans le MZ.

• SOUS-MONITEUR

Une commande du moniteur est prévue pour l'entrée de données en mémoire, il s'agit de « M ». Tapez MC000 puis la touche CR. Vous verrez alors apparaître le numéro d'adresse demandée suivi d'une valeur HEXA 8 bits correspondant au contenu actuel de l'adresse mémoire C000, puis le curseur. Sans modifier l'emplacement du curseur, tapez le premier code HEXA du programme (CD) puis la touche CR. Apparaît alors le numéro d'adresse suivant suivi de son contenu et du curseur. Tapez le second code puis la touche CR... et ainsi de suite jusqu'au dernier code du programme (C0). Après un dernier CR, tapez SHIFT-BREAK. Pour vérifier vos entrées, tapez DC000 puis CR, vous aurez la liste de tous vos codes entrés. Pour lancer le programme, tapez JC000.

Pour sauvegarder ce programme, tapez :
SC000C01AC000
Filename ? TITRE

C000 correspond à l'adresse de début du programme, C01A à l'adresse de fin, le second C000, à l'adresse d'éxécution. Ce programme s'auto-éxécutera donc automatiquement dès la fin de son chargement.

• SOUS BASIC

Un bel exemple valant mieux qu'un long discours, l'étude du programme basic 4 vous fournit toutes les indications pour entrer tout programme LM sous Basic. Attention toutefois que ce programme LM n'empiète pas dans la zone du Basic, les effets seraient alors surprenants ! Pour récupérer le Basic après l'exécution du programme LM, taper la touche CTRL puis simultanément le bouton RESET. (Uniquement pour MZ 700 et 800).

• A L'AIDE D'UN ASSEMBLEUR

L'assembleur, c'est l'outil idéal pour programmer LM. Les notices des différents assembleurs étant suffisamment explicites, il vous sera très facile d'entrer ce programme.

• A L'AIDE DE SAISIE LM

Aucun problème pour entrer et sauvegarder notre programme à l'aide de SAISIE LM paru dans notre N°12 puisqu'il démarre en C000 qui est une adresse admissible par cet excellent utilitaire de saisie.

Quelques remarques pour conclure. Ce programme démontre, s'il le fallait encore la fulgurante rapidité du Langage machine, son avarice chronique en taille mémoire (26 octets) et sa supériorité, dans certains domaines (commandes conditionnelles) à un langage dit évolué comme le Basic. L'auteur utilise quotidiennement ce langage de programmation et semble être atteint, aux dires de ses proches, d'une certaine déformation, commune aux mordus de l'octet, tendant à considérer certaines tournures de programmation comme des évidences alors qu'elles représentent un casse-tête insurmontable pour le néophyte. C'est pourquoi il est possible que certains points de ces leçons aient été mal expliquées ou insuffisamment développés. N'hésitez pas, dans ce cas, à nous le faire savoir pour que nous revenions sur ces détails et améliorons le contenu de nos prochaines leçons.

S. BIZOIRRE

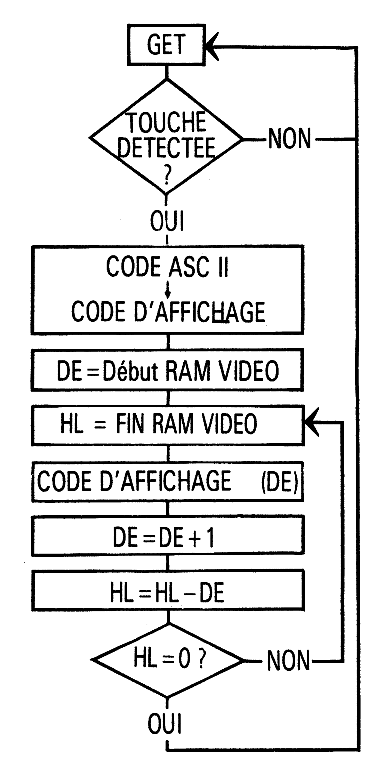


Figure 1

```
10 GET A$
20 IF A$="" GOTO 10
30 A=ASC(A$)
40 DE=$D000
50 HL=$D3E8
60 POKE DE,A
70 DE=DE+1
80 HL=HL-DE
90 IF HL<>0 GOTO 50
100 GOTO 10
```

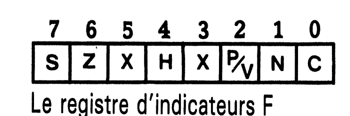
Figure 2

```
1 ' C000 CD 1B 00   CALL 001BH
2 ' C003 FE 00      CP   00H
3 ' C005 CA 00 C0   JP   Z,C000H
4 ' C008 CD B9 0B   CALL 0BB9H
5 ' C00B 11 00 D0   LD   DE,D000H
6 ' C00E 21 E8 D3   LD   HL,D3E8H
7 ' C011 12         LD   (DE),A
8 ' C012 13         INC  DE
9 ' C013 ED 52      SBC  HL,DE
10 'C015 C2 0E C0   JP   NZ,C00EH
11 'C018 C3 00 C0   JP   C000H
```

Figure 3

```
10 FORI=$C000 TO $C01A
20 READ A
30 POKE I,A
40 NEXT I:END
50 DATA $CD,$1B,$00,$FE,$00,$CA,$00
60 DATA $C0,$CD,$B9,$0B,$11,$00,$D0
70 DATA $21,$E8,$D3,$12,$13,$ED,$52
80 DATA $C2,$0E,$C0,$C3,$00,$C0
```

Figure 4



Le registre d'indicateurs F

C bit de retenue (CARRY)
N ajustement décimal ADD et SUB
P/V parité, dépassement de capacité (Parity overflow)
H demi-retenues (Half carry)
Z bit de test zéro
S bit de signe
X non utilisé

DESASSEMBLEUR

Un désassembleur convertit en mnémoniques officiels Z80 les codes lus en mémoire et vous permet ainsi de contrôler ou d'étudier un programme en Lm écrit en mémoire. Cette version due à un Sharpentier d'élite, Monsieur PERRET demande l'adresse de départ du désassemblage en HEXA puis affiche, page par page, les codes contenus dans cette portion de mémoire.

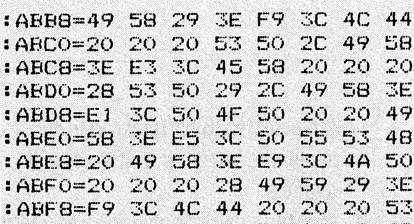
Un appui sur CR passe à la page suivante, un appui sur espace provoque la demande d'une nouvelle adresse de désassemblage. Introduisez SAISIE LM et retenez votre souffle car l'idéal, pour éviter les sauvegardes intermédiaires serait que vous saisissiez ce programme en une seule fois. Dès le dernier octet entré, vérifiez très soigneusement chaque ligne et effectuez une sauvegarde sur cassette. Le plus beau programme LM que vous puissiez posséder étant le Basic ; pour l'étudier, il vous suffira d'entrer votre Basic, puis de taper :

LIMIT $ A000
LOAD « DESASSEMBLEUR »
USR ($A000)

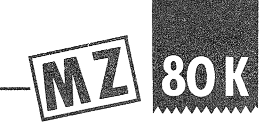
Il ne vous restera plus qu'à découvrir les commandes que nous avons déjà étudiées.

```
:A000=D3 E0 21 00 00 11 00 C0
:A008=01 00 10 ED B0 D3 E4 21
:A010=00 00 11 00 B0 01 00 10
:A018=ED B0 D3 E0 21 00 B0 11
:A020=00 00 01 00 10 ED B0 3E
:A028=02 32 7E 00 32 49 0E 32
:A030=87 0E 32 DE 0E 31 FF BF
:A038=21 46 A0 11 98 01 01 0C
:A040=00 00 00 C3 A9 A1 AF D3
:A048=FE 0E 01 CD B6 01 3E 80
:A050=D3 FE 21 7C A0 11 7B 19
:A058=01 17 00 ED B0 3E E0 32
:A060=E1 13 32 F0 13 3E FF 32
:A068=1A 1A 3E 50 32 78 13 3E
:A070=1C 32 7E 13 3E 19 32 B0
:A078=13 C3 00 12 FE 0C 20 04
:A080=3E 0D 18 18 FE 60 38 14
:A088=CD B9 0B D6 20 FE 80 38
:A090=02 3E 2E 3E 1B CD 8F 01
:A098=3E 30 C3 8F 01 3E FF D3
:A0A0=FE C9 44 45 53 41 53 53
:A0A8=45 4D 42 4C 45 55 52 20
:A0B0=20 20 20 20 20 20 20 20
:A0B8=20 43 4C 41 55 44 45 20
:A0C0=50 45 52 52 45 54 0D 28
:A0C8=49 29 4D 50 52 49 4D 41
:A0D0=4E 54 45 20 3F 3A 0D 41
:A0D8=44 52 45 53 53 45 3A 0D
:A0E0=53 53 41 44 52 45 53 53
:A0E8=45 3A 30 37 34 38 20 30
:A0F0=30 20 20 20 20 20 20 20
:A0F8=20 20 4E 4F 50 20 20 20
:A100=20 20 20 20 20 20 20 20
:A108=20 20 20 20 20 2E 20 20
:A110=20 20 0D 20 20 20 20 0D
:A118=80 A1 48 07 0D 90 09 00
:A120=00 00 F1 A0 FA A0 FF A0
:A128=0E A1 A0 04 A1 13 A1 01
:A130=F5 1F 1F 1F 1F E6 0F CD
:A138=DA 03 12 13 F1 F5 E6 0F
:A140=CD DA 03 12 13 F1 C9 D5
:A148=F5 ED 5B 22 A1 CD 30 A1
:A150=ED 53 22 A1 ED 5B 28 A1
:A158=FE 20 D2 76 A1 C3 71 A1
:A160=B7 28 12 F1 FE 60 38 0E
:A168=CD B9 0B D6 20 FE 80 38
:A170=05 3E 2E 18 01 F1 ED 5B
:A178=28 A1 12 13 ED 53 28 A1
:A180=F1 D1 C9 ED B1 37 E0 F5
:A188=7E FE 3C 28 03 F1 18 F3
:A190=23 7E FE 3E C2 9B A1 F1
:A198=37 3F C9 12 13 18 F1 7C
:A1A0=CD 30 A1 7D CD 30 A1 13
:A1A8=C9 3E 16 CD 12 00 11 A2
:A1B0=A0 CD 18 00 CD 09 00 AF
:A1B8=32 1E A1 11 C7 A0 CD 18
:A1C0=00 CD B3 09 FE 09 20 09
:A1C8=32 1E A1 00 00 00 00 00
:A1D0=00 CD 09 00 11 D7 A0 CD
:A1D8=18 00 11 E2 A0 CD 03 00
:A1E0=3A EA A0 FE 0D 28 0B 11
:A1E8=EA A0 CD 10 04 22 1A A1
:A1F0=38 B7 3E 00 32 1C A1 11
:A1F8=EA A0 06 28 3E 20 12 13
:A200=10 FC 3E 0D 12 11 FA A0
:A208=ED 53 24 A1 11 FF A0 ED
:A210=53 26 A1 11 EF A0 ED 53
:A218=22 A1 11 0D A1 ED 53 28
:A220=A1 3E 00 32 21 A1 32 20
:A228=A1 2A 1A A1 11 EA A0 CD
:A230=9F A1 2A 1A A1 7E 32 1F
:A238=A1 CD 47 A1 ED 5B 24 A1
:A240=FE ED 28 1A FE FD 28 1B
:A248=FE DD 28 17 FE CB 28 0E
:A250=3A 20 A1 B7 28 21 FE ED
:A258=CA 57 A7 C3 B6 A6 32 20
:A260=A1 18 0B 32 21 A1 3A 20
:A268=A1 FE ED CA 57 A7 2A 1A
:A270=A1 23 22 1A A1 18 BB 3A
:A278=1F A1 FE 40 DA 2F A3 FE
:A280=80 DA B8 A4 FE C0 DA 6A
:A288=A5 C3 95 A5 2A 1A A1 D5
:A290=23 22 1A A1 7E CD 47 A1
:A298=CB 7F 28 01 25 16 00 5F
:A2A0=23 19 D1 CD 9F A1 18 00
:A2A8=11 EA A0 CD 18 00 3A 1E
:A2B0=A1 B7 28 10 11 EA A0 1A
:A2B8=F5 CD 8F 01 13 F1 FE 0D
:A2C0=20 F5 28 04 3A 1C A1 3C
:A2C8=32 1C A1 2A 1A A1 23 22
:A2D0=1A A1 FE 18 C2 F7 A1 CD
:A2D8=B3 09 FE CB CA A9 A1 B7
:A2E0=C2 F2 A1 3E 16 CD 12 00
:A2E8=C3 D1 A1 21 8E A9 01 14
:A2F0=00 3A 21 A1 B7 28 0D FE
:A2F8=DD 21 A2 A9 01 05 00 28
:A300=03 21 A7 A9 3A 1F A1 E6
:A308=F0 CD 83 A1 38 1B 3E 2C
:A310=12 13 CD 91 A3 C3 A8 A2
:A318=2A 1A A1 2B 22 1A A1 3E
:A320=3F ED 5B 24 A1 12 13 12
:A328=C9 CD 18 A3 C3 A8 A2 21
:A330=58 A8 01 7E 00 CD 83 A1
:A338=D2 A8 A2 3A 1F A1 21 D6
:A340=A8 01 3A 00 CD 83 A1 D2
:A348=8C A2 21 10 A9 01 29 00
:A350=3A 21 A1 B7 28 0D FE DD
:A358=21 39 A9 01 15 00 28 03
:A360=21 4E A9 3A 1F A1 CD 83
:A368=A1 38 42 CD 91 A3 1B 21
:A370=63 A9 01 15 00 3A 21 A1
:A378=B7 28 0D FE DD 21 78 A9
:A380=01 0B 00 28 03 21 83 A9
:A388=3A 1F A1 CD 83 A1 C3 A8
:A390=A2 D5 2A 1A A1 23 22 1A
:A398=A1 7E 5F CD 47 A1 23 22
:A3A0=1A A1 7E 57 CD 47 A1 EB
:A3A8=D1 CD 9F A1 C9 3A 21 A1
:A3B0=B7 28 17 FE DD 3A 1F A1
:A3B8=21 04 AA 01 0B 00 28 03
:A3C0=21 0F AA E6 0F CD 83 A1
:A3C8=30 11 21 AC A9 01 58 00
:A3D0=3A 1F A1 E6 0F CD 83 A1
:A3D8=DA 29 A3 13 FE 01 CA EB
:A3E0=A2 FE 09 CA 86 A4 E6 07
:A3E8=FE 03 CA 5F A4 FE 06 DA
:A3F0=43 A4 18 00 CD 13 A4 DA
:A3F8=29 A3 3E 2C 12 13 CD 04
:A400=A4 C3 A8 A2 2A 1A A1 23
:A408=22 1A A1 7E CD 47 A1 CD
:A410=30 A1 C9 3A 21 A1 B7 28
:A418=0F FE DD 21 2D AA 01 07
```

```
:A420=00 28 29 21 34 AA 1B 24
:A428=3A 1F A1 E6 0F 21 1A AA
:A430=01 13 00 FE 0B 38 03 21
:A438=3B AA 3A 1F A1 E6 F0 CD
:A440=83 A1 C9 CD 13 A4 DA 29
:A448=A3 C3 A8 A2 3A 1F A1 E6
:A450=F0 CD 83 A1 D8 CD 04 A4
:A458=3E 29 12 13 37 3F C9 21
:A460=8E A9 01 1E 00 3A 21 A1
:A468=B7 28 0D FE DD 21 A2 A9
:A470=01 05 00 28 03 21 A7 A9
:A478=3A 1F A1 E6 F0 CD 83 A1
:A480=DA 29 A3 C3 A8 A2 3A 21
:A488=A1 B7 28 1A FE DD 21 A2
:A490=A9 01 05 00 28 03 21 A7
:A498=A9 3A 1F A1 E6 F0 1B CD
:A4A0=83 A1 D2 A8 A2 13 21 8E
:A4A8=A9 01 14 00 3A 1F A1 1B
:A4B0=E6 F0 CD 83 A1 C3 A8 A2
:A4B8=3A 1F A1 21 4B AA 01 0E
:A4C0=00 FE 76 20 10 3A 21 A1
:A4C8=B7 C2 29 A3 3A 1F A1 CD
:A4D0=83 A1 C3 A8 A2 3A 21 A1
:A4D8=B7 28 44 3A 1F A1 E6 F0
:A4E0=FE 70 20 31 3A 1F A1 E6
:A4E8=0F FE 0E 28 32 FE 08 38
:A4F0=03 C3 29 A3 3E 40 21 4B
:A4F8=AA 01 0E 00 CD 83 A1 3A
:A500=1F A1 D6 40 32 1F A1 13
:A508=CD 13 A4 DA 29 A3 3E 2C
:A510=12 13 C3 57 A5 3A 1F A1
:A518=E6 07 FE 06 C2 29 A3 3E
:A520=40 21 4B AA 01 0E 00 CD
:A528=83 A1 21 3B AA 01 10 00
:A530=3A 1F A1 CB 5F 20 06 21
:A538=1A AA 01 13 00 E6 F0 D6
:A540=40 13 CD 83 A1 3E 2C 12
:A548=13 3A 21 A1 B7 28 08 3E
:A550=30 32 1F A1 C3 43 A4 3A
:A558=1F A1 E6 07 21 59 AA 01
:A560=23 00 E6 07 CD 83 A1 C3
:A568=A8 A2 3A 21 A1 B7 28 0A
:A570=3A 1F A1 E6 07 FE 06 C2
:A578=29 A3 3A 1F A1 CB 5F 21
:A580=89 AA 01 22 00 28 06 21
:A588=AB AA 01 24 00 E6 F0 CD
:A590=83 A1 C3 49 A5 21 65 AB
:A598=01 C3 00 3A 21 A1 B7 28
:A5A0=19 FE DD 21 B0 AB 01 3C
:A5A8=00 28 03 21 EC AB 3A 1F
:A5B0=A1 CD 83 A1 DA 29 A3 C3
:A5B8=A8 A2 3A 1F A1 CD 83 A1
:A5C0=D2 A8 A2 21 28 AC 01 0E
:A5C8=00 CD 83 A1 CA 5A A6 21
:A5D0=36 AC 01 14 00 CD 83 A1
:A5D8=CA 61 A6 E6 0F 21 CF AA
:A5E0=01 46 00 CD 83 A1 13 3A
:A5E8=1F A1 E6 07 CA 13 A6 FE
:A5F0=02 CA 3F A6 FE 04 CA 3F
:A5F8=A6 FE 06 CA 86 A6 FE 07
:A600=CA A7 A6 3A 1F A1 E6 0F
:A608=FE 01 28 22 FE 05 28 1E
:A610=C3 A8 A2 21 15 AB 01 13
:A618=00 3A 1F A1 CB 5F 28 06
:A620=21 28 AB 01 11 00 E6 F0
:A628=CD 83 A1 C3 A8 A2 21 56
:A630=AC 01 14 00 3A 1F A1 E6
:A638=F0 CD 83 A1 C3 A8 A2 21
:A640=15 AB 01 13 00 3A 1F A1
:A648=CB 5F 28 06 21 28 AB 01
:A650=11 00 E6 F0 CD 83 A1 3E
:A658=2C 12 13 CD 91 A3 C3 A8
:A660=A2 CD 04 A4 3A 1F A1 21
:A668=4A AC 01 0C 00 CD 83 A1
:A670=C3 A8 A2 21 15 AB CB 5F
:A678=01 13 00 28 D5 21 28 AB
:A680=01 11 00 C3 52 A6 21 89
:A688=AA 01 22 00 3A 1F A1 CB
:A690=5F 28 06 21 AB AA 01 24
:A698=00 E6 F0 CB B7 1B CD 83
:A6A0=A1 CD 04 A4 C3 A8 A2 3A
:A6A8=1F A1 21 39 AB 01 2C 00
:A6B0=CD 83 A1 C3 A8 A2 3A 21
:A6B8=A1 B7 28 25 3A 1F A1 32
:A6C0=1D A1 2A 1A A1 23 22 1A
:A6C8=A1 7E 32 1F A1 CD 47 A1
:A6D0=E6 07 FE 06 28 0B 2A 1A
:A6D8=A1 2B 2B 22 1A A1 C3 29
:A6E0=A3 3A 1F A1 FE 40 30 44
:A6E8=21 6A AC 01 11 00 CB 5F
:A6F0=28 06 21 7B AC 01 17 00
:A6F8=E6 F0 CD 83 A1 DA 29 A3
:A700=ED 5B 26 A1 3A 21 A1 B7
:A708=CA 57 A5 FE DD 21 2D AA
:A710=01 07 00 28 03 21 34 AA
:A718=3E 30 CD 83 A1 3A 1D A1
:A720=CD 30 A1 3E 29 12 C3 A8
:A728=A2 C3 49 A5 21 92 AC 01
:A730=12 00 E6 C0 CD 83 A1 3A
:A738=1F A1 21 A4 AC 01 10 00
:A740=CB 5F 28 03 21 B4 AC ED
:A748=5B 26 A1 E6 30 CD 83 A1
:A750=3E 2C 12 13 C3 04 A7 3A
:A758=1F A1 FE 40 DA 29 A3 FE
:A760=80 30 11 E6 07 28 1F FE
:A768=01 28 59 FE 02 28 69 FE
:A770=03 CA F9 A7 21 FE AC 01
:A778=CC 00 3A 1F A1 CD 83 A1
:A780=D2 A8 A2 C3 29 A3 3E 00
:A788=21 C4 AC 01 11 00 CD 83
:A790=A1 ED 5B 26 A1 CD A9 A7
:A798=DA 29 A3 21 D5 AC 01 07
:A7A0=00 3E 00 CD 83 A1 C3 A8
:A7A8=A2 21 F2 AC 01 0C 00 3A
:A7B0=1F A1 CB 5F 28 08 21 3B
:A7B8=AA 01 10 00 D6 40 E6 F0
:A7C0=CD 83 A1 C9 3E 01 21 C4
:A7C8=AC 01 11 00 CD 83 A1 CD
:A7D0=A9 A7 DA 29 A3 C3 A8 A2
:A7D8=3A 1F A1 21 DC AC 01 16
:A7E0=00 E6 0F CD 83 A1 3A 1F
:A7E8=A1 D6 40 E6 F0 21 8E A9
:A7F0=01 14 00 CD 83 A1 C3 A8
:A7F8=A2 3A 1F A1 E6 F0 FE 60
:A800=CA 29 A3 3E 40 21 4B AA
:A808=01 0E 00 CD 83 A1 ED 5B
:A810=26 A1 3A 1F A1 CB 5F 20
:A818=20 3E 28 12 13 CD 91 A3
:A820=1B 3E 29 12 13 3E 2C 12
:A828=13 21 8E A9 01 14 00 3A
:A830=1F A1 D6 40 E6 F0 C3 E6
:A838=A7 21 8E A9 01 14 00 D6
:A840=40 E6 F0 CD 83 A1 3E 2C
:A848=12 13 3E 28 12 13 CD 91
:A850=A3 1B 3E 29 12 C3 A8 A2
:A858=00 3C 4E 4F 50 3E 08 3C
:A860=45 58 20 20 20 41 46 2C
:A868=41 46 27 3E 02 3C 4C 44
:A870=20 20 20 28 42 43 29 2C
:A878=41 3E 0A 3C 4C 44 20 20
:A880=20 41 2C 28 42 43 29 3E
:A888=12 3C 4C 44 20 20 20 28
:A890=44 45 29 2C 41 3E 1A 3C
:A898=4C 44 20 20 20 41 2C 28
:A8A0=44 45 29 3E 07 3C 52 4C
:A8A8=43 41 3E 17 3C 52 4C 41
:A8B0=3E 27 3C 44 41 41 3E 37
:A8B8=3C 53 43 46 3E 0F 3C 52
:A8C0=52 43 41 3E 1F 3C 52 52
:A8C8=41 3E 2F 3C 43 50 4C 3E
:A8D0=3F 3C 43 43 46 3E 10 3C
:A8D8=44 4A 4E 5A 20 3E 20 3C
:A8E0=4A 52 20 20 20 4E 5A 2C
:A8E8=3E 30 3C 4A 52 20 20 20
:A8F0=4E 43 2C 3E 18 3C 4A 52
:A8F8=20 20 20 3E 28 3C 4A 52
:A900=20 20 20 5A 2C 3E 38 3C
:A908=4A 52 20 20 20 43 2C 3E
:A910=22 3C 4C 44 20 20 20 28
:A918=3E 32 3C 4C 44 20 20 20
:A920=28 3E 2A 3C 4C 44 20 20
:A928=20 48 4C 2C 28 3E 3A 3C
:A930=4C 44 20 20 20 41 2C 28
:A938=3E 22 3C 4C 44 20 20 20
:A940=28 3E 2A 3C 4C 44 20 20
:A948=20 49 58 2C 28 3E 22 3C
:A950=4C 44 20 20 20 28 3E 2A
:A958=3C 4C 44 20 20 20 49 59
:A960=2C 28 3E 22 3C 29 2C 48
:A968=4C 3E 32 3C 29 2C 41 3E
:A970=2A 3C 29 3E 3A 3C 29 3E
:A978=22 3C 29 2C 49 58 3E 2A
:A980=3C 29 3E 22 3C 29 2C 49
:A988=59 3E 2A 3C 29 3E 00 3C
:A990=42 43 3E 10 3C 44 45 3E
:A998=20 3C 48 4C 3E 30 3C 53
:A9A0=50 3E 20 3C 49 58 3E 20
:A9A8=3C 49 59 3E 01 3C 4C 44
:A9B0=20 20 3E 02 3C 4C 44 20
:A9B8=20 3E 03 3C 49 4E 43 20
:A9C0=3E 04 3C 49 4E 43 20 3E
:A9C8=05 3C 44 45 43 20 3E 06
:A9D0=3C 4C 44 20 20 3E 09 3C
:A9D8=41 44 44 20 20 48 4C 2C
:A9E0=3E 0A 3C 4C 44 20 20 3E
:A9E8=0B 3C 44 45 43 20 3E 0C
:A9F0=3C 49 4E 43 20 3E 0D 3C
:A9F8=44 45 43 20 3E 0E 3C 4C
:AA00=44 20 20 3E 09 3C 41 44
:AA08=44 20 20 49 58 2C 3E 09
:AA10=3C 41 44 44 20 20 49 59
:AA18=2C 3E 00 3C 42 3E 10 3C
:AA20=44 3E 20 3C 48 3E 30 3C
:AA28=28 48 4C 29 3E 30 3C 28
:AA30=49 58 2B 3E 30 3C 28 49
:AA38=59 2B 3E 00 3C 43 3E 10
:AA40=3C 45 3E 20 3C 4C 3E 30
:AA48=3C 41 3E 40 3C 4C 44 20
:AA50=20 3E 76 3C 48 41 4C 54
:AA58=3E 00 3C 42 3E 01 3C 43
:AA60=3E 02 3C 44 3E 03 3C 45
:AA68=3E 04 3C 48 3E 05 3C 4C
:AA70=3E 06 3C 28 48 4C 29 3E
:AA78=07 3C 41 3E 06 3C 49 58
:AA80=2B 3E 06 3C 28 49 59 2B
:AA88=3E 80 3C 41 44 44 20 20
:AA90=41 2C 3E 90 3C 53 55 42
:AA98=20 20 3E A0 3C 41 4E 44
:AAA0=20 20 3E B0 3C 4F 52 20
:AAA8=20 20 3E 80 3C 41 44 43
:AAB0=20 20 41 2C 3E 90 3C 53
:AAB8=42 43 20 20 41 2C 3E A0
:AAC0=3C 58 4F 52 20 20 3E B0
:AAC8=3C 43 50 20 20 20 3E 00
:AAD0=3C 52 45 54 20 3E 08 3C
:AAD8=52 45 54 20 3E 01 3C 50
:AAE0=4F 50 20 3E 02 3C 4A 50
:AAE8=20 20 3E 0A 3C 4A 50 20
:AAF0=20 3E 04 3C 43 41 4C 4C
:AAF8=3E 0C 3C 43 41 4C 4C 3E
:AB00=05 3C 50 55 53 48 3E 07
:AB08=3C 52 53 54 20 3E 0F 3C
:AB10=52 53 54 20 3E C0 3C 4E
:AB18=5A 3E D0 3C 4E 43 3E E0
:AB20=3C 50 4F 3E F0 3C 50 3E
:AB28=C0 3C 5A 3E D0 3C 43 3E
:AB30=E0 3C 50 45 3E F0 3C 4D
:AB38=3E C7 3C 30 3E D0 3C 31
:AB40=30 48 3E E7 3C 32 30 48
:AB48=3E F7 3C 33 30 48 3E CF
:AB50=3C 38 3E DF 3C 31 38 48
:AB58=3E EF 3C 32 38 48 3E FF
:AB60=3C 33 38 48 3E C9 3C 52
:AB68=45 54 3E D9 3C 45 58 58
:AB70=3E E9 3C 4A 50 20 20 20
:AB78=28 48 4C 29 3E F9 3C 4C
:AB80=44 20 20 20 53 50 2C 48
:AB88=4C 3E E3 3C 45 58 20 20
:AB90=20 28 53 50 29 2C 48 4C
:AB98=3E F3 3C 44 49 3E EB 3C
:ABA0=45 58 20 20 20 44 45 2C
:ABA8=48 4C 3E FB 3C 45 49 3E
:ABB0=E9 3C 4A 50 20 20 20 28
```



```
:ABB8=49 58 29 3E F9 3C 4C 44
:ABC0=20 20 20 53 50 2C 49 58
:ABC8=3E E3 3C 45 58 20 20 20
:ABD0=28 53 50 29 2C 49 58 3E
:ABD8=E1 3C 50 4F 50 20 20 49
:ABE0=58 3E E5 3C 50 55 53 48
:ABE8=20 49 58 3E E9 3C 4A 50
:ABF0=20 20 20 28 49 59 29 3E
:ABF8=F9 3C 4C 44 20 20 20 53
:AC00=50 2C 49 59 3E E3 3C 45
:AC08=58 20 20 20 28 53 50 29
:AC10=2C 49 59 3E E1 3C 50 4F
:AC18=50 20 20 49 59 3E E5 3C
:AC20=50 55 53 48 20 49 59 3E
:AC28=C3 3C 4A 50 20 20 3E CD
:AC30=3C 43 41 4C 4C 3E D3 3C
:AC38=4F 55 54 20 20 28 3E DB
:AC40=3C 49 4E 20 20 20 41 2C
:AC48=28 3E D3 3C 48 29 2C 41
:AC50=3E DB 3C 48 29 3E C0 3C
:AC58=42 43 3E D0 3C 44 45 3E
:AC60=E0 3C 48 4C 3E F0 3C 41
:AC68=46 3E 00 3C 52 4C 43 3E
:AC70=10 3C 52.4C 3E 20 3C 53
:AC78=4C 41 3E 00 3C 52 52 43
:AC80=3E 10 3C 52 52 3E 20 3C
:AC88=53 52 41 3E 30 3C 53 52
:AC90=4C 3E 40 3C 42 49 54 3E
:AC98=80 3C 52 45 53 3E C0 3C
:ACA0=53 45 54 3E 00 3C 30 3E
:ACA8=10 3C 32 3E 20 3C 34 3E
:ACB0=30 3C 36 3E 00 3C 31 3E
:ACB8=10 3C 33 3E 20 3C 35 3E
:ACC0=30 3C 37 3E 00 3C 49 4E
:ACC8=3E 01 3C 4F 55 54 20 20
:ACD0=28 43 29 2C 3E 00 3C 2C
:ACD8=28 43 29 3E 02 3C 53 42
:ACE0=43 20 20 48 4C 2C 3E 0A
:ACE8=3C 41 44 43 20 20 48 4C
:ACF0=2C 3E 40 3C 42 3E 50 3C
:ACF8=44 3E 60 3C 48 3E 44 3C
:AD00=4E 45 47 3E 45 3C 52 45
:AD08=54 4E 3E 4D 3C 52 45 54
:AD10=49 3E 46 3C 49 4D 30 3E
:AD18=56 3C 49 4D 31 3E 5E 3C
:AD20=49 4D 32 3E 47 3C 4C 44
:AD28=20 20 20 49 2C 41 3E 57
:AD30=3C 4C 44 20 20 20 41 2C
:AD38=49 3E 4F 3C 4C 44 20 20
:AD40=20 52 2C 41 3E 5F 3C 4C
:AD48=44 20 20 20 41 2C 52 3E
:AD50=6F 3C 52 4C 44 3E A3 3C
:AD58=4F 55 54 49 3E AB 3C 4F
:AD60=55 54 44 3E 06 E0 63 FB
:AD68=9D 56 64 63 23 62 00 00
:AD70=4F 54 49 52 3E BB 3C 4F
:AD78=54 44 52 3E A0 3C 4C 44
:AD80=49 3E B0 3C 4C 44 49 52
:AD88=3E A8 3C 4C 44 44 3E B8
:AD90=3C 4C 44 44 52 3E A1 3C
:AD98=43 50 49 3E B1 3C 43 50
:ADA0=49 52 3E A9 3C 43 50 44
:ADA8=3E B9 3C 43 50 44 52 3E
:ADB0=A2 3C 49 4E 49 3E B2 3C
:ADB8=49 4E 49 52 3E AA 3C 49
:ADC0=4E 44 3E BA 3C 49 4E 44
:ADC8=52 3E 00 4E 44 52 3E 00
```



LANGAGE PILOT
(S BASIC 3 ou 4)

Ce programme est un interpréteur Pilot écrit en Super-basic pour le Sharp MZ 80 K. Il a été inspiré par des programmes similaires publiés par le club anglais Sharpsoft Ltd et la revue Creative Computing. Son écriture en Basic permettra à l'utilisateur de le modifier, de l'améliorer ou de changer la syntaxe des instructions s'il le désire.
Ce Pilot est écrit sous forme structurée (sauf pour les 4 premières lignes qui sont des « astuces de programmation » permettant le passage de variables numériques entre le Pilot et le Basic et qui ne devront en aucun cas être modifiées sous peine de voir l'interpréteur Pilot et éventuellement le Basic lui-même, se comporter de façon aberrante !).

Lors du chargement de l'interprétateur, le titre s'affiche suivi de la taille mémoire disponible puis du symbole d'attente > suivi du curseur clignotant.

L'interprétateur Pilot est alors à la disposition de l'utilisateur. Celui-ci peut alors entrer soit une commande directe, soit une ligne de programme Pilot commençant par un numéro de ligne compris entre 0 et 255 (ceci parce que les lignes du programme Pilot sont stockées dans le tableau P§). Toute ligne commençant par un numéro de ligne supérieur à 255 provoquera une erreur.

Une remarque importante :
L'entrée de ces lignes de programme Pilot se fait en réponse à un INPUT et donc l'utilisation de la virgule fera perdre une partie de la ligne entrée.

L'entrée des lignes Pilot suit les règles classiques du Basic, c'est-à-dire (par exemple) que pour effacer une ligne, il suffit d'entrer son numéro suivi de CR.

Les commandes directes sont les suivantes :

- ! CR effacera tout l'écran et remettra le curseur en position Home.
- H CR affichera la liste des commandes directes (HELP)
- R CR lance l'exécution du programme Pilot à sa première ligne (RUN)
- N CR efface tout le programme. Pour éviter tout fause manœuvre, l'interprétateur demande confirmation de cette commande (NEW)
- E CR redonne le contrôle au Basic (EXIT)
- S CR enregistre un programme Pilot sur cassette (le titre doit avoir au plus 10 caractères). (SAVE)
- G CR charge en mémoire le programme Pilot à partir de la cassette. Le Pilot demande le nom du programme et le charge. Dans le cas où l'on ne veut pas entrer le nom de ce programme, on entrera — en réponse à la demande du Pilot. Dans ce cas, le premier enregistrement trouvé sera chargé en mémoire. (GET).
- L CR liste de tout ou partie du programme Pilot (LIST)
- D CR efface une partie du programme (DELETE)
- P CR active ou non l'imprimante pour les listings (PRINTER)
- M CR renumérote tout le programme. L'utilisateur a le choix du nouveau numéro de la première ligne et de l'incrément sur les numéros. Cependant, si par suite de la renumérotation, certaines

lignes devaient avoir un numéro plus grand que 255, elles seront perdues (RENUM).

A CR numérote automatiquement les lignes lors de l'entrée ; l'utilisateur a le choix du premier numéro et de l'incrément. Cette numérotation s'arrête automatiquement dès que le numéro de ligne dépasse 255 ou que la ligne portant ce numéro existe déjà en mémoire. L'utilisateur peut arrêter cette numérotation en entrant Z Cr après l'affichage du numéro de ligne. Un signal sonore sera alors entendu (AUTO).

Toute commande invalide provoquera un signal sonore.

Lors de l'entrée des lignes de programme Pilot, une analyse syntaxique sommaire est effectuée par l'interpréteur qui ne teste que la présence de : dans la ligne. (Sur ce point des améliorations sont possibles).

Une ligne Pilot se compose au plus de 4 parties (sans compter le numéro de ligne qui n'intervient que pour le classement des lignes).

PARTIE I

Il s'agit d'un seul symbole qui indique le type d'instruction contenue dans la ligne. Les symboles autorisés seront x A C D E H J M P Q R T U V X

PARTIE II

Cette partie II qui peut (et doit, dans le cas où la partie I est x) ne pas exister est constituée soit de la lette Y (Yes) ou de la lettre N (No).

Ces lettres font référence à un drapeau interne (variable Basic MF) commandé par l'instruction M.

Lorsque la partie II de la ligne est Y, cette ligne sera exécutée si le drapeau est Yes (MF = I) ou sautée si le drapeau est No (MF = 0). Les choses sont inversées si la lettre est N.

L'absence de Y ou N conduira le Pilot à toujours exécuter la ligne.

PARTIE III»

Il s'agit de : qui est obligatoire.

PARTIE IV

On trouve à cet endroit ce que doit faire le Pilot et qui va maintenant être explicité.

x : début :
va stocker l'étiquette "début" dans le tableau des étiquettes. Ces étiquettes serviront dans les instructions de saut (J) et d'appel de sous-programmes (U). Par exemple : JY : début fera poursuivre l'exécution du programme à la ligne référencée par l'étiquette "début" (si le drapeau est Y).
Les tableaux de stockage sont L et L§. Dans l'interpréteur initial, ces tableaux sont définis par DIM L(20), L§(20) ce qui doit être suffisant dans la pratique. Cependant l'utilisateur peut très bien modifier ces dimensions. Dans le cas où plusieurs lignes portent la même étiquette, seule la première ligne pourra être atteinte par un Jump (J) ou un saut à un sous-programme (U).

J : début
cette instruction est, comme nous venons de le voir, l'instruction de saut à la ligne x : début.

R : ceci est commentaire
Une telle ligne est toujours sautée par l'interpréteur. Elle ne sert qu'à contenir des commentaires.

Q :
Quit est l'instruction d'arrêt du programme Pilot et de l'interpréteur Pilot. Le contrôle est rendu au Basic. Il s'agit donc d'un retour au Basic, à partir d'un programme Pilot. En temps normal, un programme Pilot s'arrête de lui-même lorsqu'il n'y a plus de lignes à exécuter et on retrouve alors le « mode éditeur ».

U : étiquette
poursuit l'exécution du programme au sous-programme commençant par la ligne x : étiquette. La rencontre de l'instruction E : fera revenir au programme principal. On peut imbriquer jusqu'à 20 sous-programmes (la pile est le tableau ST et le pointeur de pile est SP).

A : ...
est le Input du Pilot (instruction Accept). Cette instruction est utilisée de trois façons différentes.

A : stocke la donnée entrée au clavier dans la variable Basic B§ pour utilisation éventuelle dans l'instruction M.

A:N§ stocke la donnée entrée au clavier dans la variable Basic B§ et cette fois dans la variable Pilot N§ (ces variables sont stockées dans le tableau V§). Bien remarquer que seule une lettre peut être utilisée avant §.

A:X# stocke la valeur numérique entrée au clavier dans la variable Basic et Pilot X. En effet, l'utilisateur pourra remarquer que l'interpréteur n'utilise aucune variable Basic dont le nom est constitué d'une seule lettre. Cela permet donc de faire coïncider ces variables Basic et Pilot. Le passage des valeurs entre Pilot et Basic se fait par quelques Pokes qui modifient les lignes 20 et 30 de l'interpréteur et par appel des sous programmes des lignes 20 et 30.
En conséquence, lors de toute modification du Pilot, il faudra bien prendre garde à ne pas modifier le début du programme (le chiffre de la ligne 10 peut être changé mais il doit toujours être constitué de 4 caractères) et ne pas utiliser des variables Basic dont le nom ne comporte qu'une seule lettre.

M:nom
(Match) compare la dernière variable alphanumérique entrée par A (et donc stockée dans la variable Basic B§) avec la chaîne "nom". S'il y a coïncidence, le drapeau MF est mis à I et sinon, il sera mis à 0.
On peut ici faire des testes multiples en utilisant le OU logique (codé !). Exemple : M : nom1 ! nom2 ! nom 3.

C:X = expression
(Compute) est l'instruction de calcul. La variable est une variable numérique Pilot ou Basic (donc une seule lettre pour le nom) et expression est une formule Basic. Exemple : C:A = RND(10).

X:condition
est une instruction de test sur les variables numériques. La condition est écrite en basic et son résultat est traduit par le drapeau MF. Exemple : X:(C<10)

V:n1;n2
place le curseur au point de coordonnées n1,n2 (il s'agit de l'instruction Basic Cursor). Bien remarquer le point virgule entre les deux arguments.

D:n1;n2
est l'instruction Move n1,n2 du basic.

P:k;n1;n2
est l'instruction Plot k,n1,n2 du basic.

T:expression ou H:expression
impriment sur l'écran « expression » à partir de la position du curseur. T fait ensuite passer à la ligne alors que H laisse le curseur derrière ce qui vient d'être écrit.
Dans « expression » on peut utiliser des variables alphanumérique Pilot (ex:N§) ou des variables numériques (ex:C#).
Par exemple, le programme suivant :
T:Quel est votre prénom ?
A:X§
T:Bonjour X§
affichera « Quel est votre prénom ? ». L'utilisateur entre alors (par exemple) Pierre et le Pilot va alors afficher « Bonjour Pierre ».
Remarque :
T: fait passer à la ligne suivante.
T:! efface l'écran.

Voici maintenant quelques exemples de programmes en Pilot.

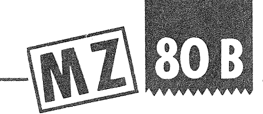
```
0 *:DEBUT
1 T:!
2 C:A=INT(RND(1000))
3 C:C=INT(RND(A))
4 T:JE CHOISIS UN NOMBRE ENTRE 0 ET A#
5 T:
6 T:A VOUS DE LE TROUVER.
7 C:I=0
8 *:RECOMMENCE
9 T:
10 H:VOTRE NOMBRE :
11 A:A#
12 C:I=I+1
13 X:(A=C)
14 T:
15 TY:BRAVO ! VOUS AVEZ TROUVE EN I# ESSAIS
16 JY:FIN
17 X:(A<C)
18 TY:TROP PETIT RECOMMENCEZ
19 JY:RECOMMENCE
20 T:TROP GRAND RECOMMENCEZ
21 J:RECOMMENCE
22 *:FIN
23 T:
24 T:VOULEZ VOUS RECOMMENCER ?
25 A:N$
26 M:OUI
27 JY:DEBUT
28 T:AU REVOIR
```

```
0 *:DEBUT
1 T:!
2 T:EQUATIONS DU SECOND DEGRE
3 T:
4 T:
5 H:Coefficient A:
6 A:A#
7 T:
8 H:Coefficient B:
9 A:B#
10 T:
11 H:Coefficient C:
12 A:C#
13 T:
14 C:D=B*B-4*A*C
15 C:E=SQR(ABS(D))/2/A
16 C:U=-B/2/A
17 X:(D>0)
18 JY:DISC.>0
19 X:(D=0)
20 JY:DISC.=0
21 T:Racines:U#+/-i.E#
22 J:FIN
23 *:DISC.=0
24 T:Racine double:U#
25 J:FIN
26 *:DISC.>0
27 C:V=U+E
28 C:W=U-E
29 T:Racine:V#
30 T:Racine:W#
31 *:FIN
32 T:
33 H:ON RECOMMENCE :
34 A:N$
35 M:OUI
36 JY:DEBUT
37 T:AU REVOIR !
```

<u>Le listing de l'interpréteur PILOT:</u>

```
10 GOTO2480
20 W=A0:RETURN
30 A0=W:RETURN
40 @"T":RETURN
50 REM ********************************
60 PROCEDURE "T"
70 IF A$="!" DO
80 ' PRINT"■"
90 ELSE DO
100 ' K0=1:J0=LPOS(A$,"$",K0)
110 ' WHILE J0>0 DO
120 '   @"VAR$":J0=LPOS(A$,"$",K0)
130 ' WEND
140 ' K0=1:J0=LPOS(A$,"#",K0)
150 ' WHILE J0>0 DO
160 '   @"VAR#":J0=LPOS(A$,"#",K0)
170 ' WEND
180 ' PRINT A$
190 ENDIF
200 RETURN
210 REM ********************************
220 PROCEDURE "H"
230 IF A$="!" DO
240 ' PRINT"■■";
250 ELSE DO
260 ' K0=1:J0=LPOS(A$,"$",K0)
270 ' WHILE J0>0 DO
280 '   @"VAR$":J0=LPOS(A$,"$",K0)
290 ' WEND
300 ' K0=1:J0=LPOS(A$,"#",K0)
310 ' WHILE J0>0 DO
320 '  @"VAR#":J0=LPOS(A$,"#",K0)
330 ' WEND
340 ' PRINT A$;
350 ENDIF
360 RETURN
370 REM ********************************
380 PROCEDURE "VAR#"
390 K1=K0+J0:V$=MID$(A$,K1-2,1)
400 V0=ASC(V$)
410 IF(V0<65)+(V0>90) DO
420 ' ER=10:A$=""
430 ELSE DO
440 ' POKE 24434,V0:GOSUB30
450 ' A$=IN$(A$,K1-3,K1,STR$(A0))
460 ENDIF
470 RETURN
480 REM ********************************
490 PROCEDURE "VAR$"
500 L0=-1:K1=K0+J0:G$=MID$(A$,K1-2,1)
510 RCPEAT
520 ' L0=L0+1:ID=(LEFT$(V$(L0),1)=G$)
530 UNTIL(ID<>0)+(L0>=V1)
540 IF ID=0 DO
550 ' A$="":ER=9
560 ELSE DO
570 ' I$=V$(L0):L2=LEN(I$)-2
580 ' A$=IN$(A$,K1-3,K1,RIGHT$(I$,L2))
590 ' K0=J0+L2-1
600 ENDIF
610 RETURN
620 REM ********************************
630 PROCEDURE "A"
640 INPUT"?";E$
650 IF A$=""DO
660 ' B$=E$
670 ELSIF LEN(A$)=2 DO
680 ' B$=E$
690 ' IF LPOS(A$,"$",1)>0 DO
700 '   IF V1>=0 DO
710 '     L0=-1
720 '     REPEAT
730 '       L0=L0+1:ID=(LEFT$(V$(L0),2)=A$)
740 '     UNTIL(ID=-1)+(L0=V1)
750 '     IF ID=-1 DO
760 '       V$(L0)=A$+E$
770 '     ELSE DO
780 '       V1=V1+1:V$(V1)=A$+E$
790 '     ENDIF
800 '   ELSE DO
810 '     V1=V1+1:V$(V1)=A$+E$
820 '   ENDIF
830 ' ELSIF LPOS(A$,"#",1)>0 DO
840 '   V$=LEFT$(A$,1):V0=ASC(V$)
850 '   IF(V0<65)+(V0>90) DO
860 '     ER=10
870 '   ELSE DO
880 '     DEF FNF(X)=EVAL(B$):A0=FNF(X)
890 '     POKE 24420,V0:GOSUB20
900 '   ENDIF
910 ' ELSE ER=10
920 ' ENDIF
930 ELSE ER=10
940 ENDIF
950 RETURN
960 REM ********************************
970 PROCEDURE "M"
980 MF=0:M1=0:D$=#"ESP"(A$)
990 L0=0:ID=0
1000 REPEAT
1010 ' L0=L0+1
1020 ' IF MID$(D$,L0,1)="!"DO
1030 '   F$=MID$(D$,M1+1,L0-M1-1)
1040 '   MF=-(F$=B$)
1050 '   IF MF=1 DO:L0=LEN(D$):ID=1
1060 '   ELSE DO:M1=L0
1070 '   ENDIF
1080 ' ENDIF
1090 UNTIL L0>=LEN(D$)
1100 IF ID=0 DO:L0=L0+1
1110 ' F$=MID$(D$,M1+1,L0-M1-1)
1120 ' MF=-(F$=B$)
1130 ENDIF
1140 RETURN
1150 REM ********************************
1160 PROCEDURE "X"
1170 DEF FNF(X)=EVAL(A$):MF=-FNF(X)
1180 RETURN
1190 REM ********************************
1200 PROCEDURE "C"
1210 K0=LPOS(A$,"=",2)
1220 IF K0=0 DO
1230 ' ER=10
1240 ELSE DO
1250 ' V$=LEFT$(A$,K0):V0=ASC(V$)
1260 ' IF(V0<65)+(V0>90) DO
1270 '   ER=10
1280 ' ELSE DO
1290 '   W$=IN$(A$,0,K0+2,"")
1300 '   DEF FNF(X)=EVAL(W$)
1310 '   A0=FNF(X)
1320 '   POKE 24420,V0:GOSUB20
1330 ' ENDIF
1340 ENDIF
1350 RETURN
1360 REM ********************************
1370 PROCEDURE "J"
1380 IF LT=-1 DO:ER=5
1390 ELSE DO
1400 ' J0=-1
1410 ' REPEAT
1420 '   J0=J0+1
1430 ' UNTIL(A$=L$(J0))+(J0=LT)
1440 ' IF A$<>L$(J0) DO:ER=5
1450 ' ELSE DO:I0=L(J0)-1
1460 ' ENDIF
1470 ENDIF
1480 RETURN
1490 REM ********************************
1500 PROCEDURE "U"
1510 IF SP=20 DO:ER=6
1520 ELSE DO
1530 ' ST(SP)=I0:SP=SP+1
1540 ' IF LT=-1 DO:ER=5
1550 ' ELSE DO
1560 '   J0=-1
1570 '   REPEAT
1580 '     J0=J0+1
1590 '   UNTIL(A$=L$(J0))+(J0=LT)
1600 '   IF A$<>L$(J0) DO:ER=5
1610 '   ELSE DO:I0=L(J0)-1
1620 '   ENDIF
1630 ' ENDIF
1640 ENDIF
1650 RETURN
1660 REM ********************************
1670 PROCEDURE "E"
1680 IF SP>0 DO
1690 ' SP=SP-1:I0=ST(SP)
1700 ELSE DO
1710 ' ER=7
1720 ENDIF
1730 RETURN
1740 REM ********************************
1750 PROCEDURE "V"
1760 K0=LPOS(A$,";",1)
1770 IF K0=0 DO
1780 ' Y0=0:X0=0
1790 ELSE DO
1800 ' DEF FNF(X)=EVAL(MID$(A$,1,K0-1))
```

```
1810 ' X0=FNF(X)
1820 ' DEF FNF(X)=EVAL(IN$(A$,0,K0+1,""))
1830 ' Y0=FNF(X)
1840 ENDIF
1850 X0=ABS(X0):Y0=ABS(Y0)
1860 IF(X0>39)+(Y0>24)DO:ER=8
1870 ELSE CURSOR X0,Y0
1900 REM ********************************
1910 PROCEDURE "D"
1920 K0=LPOS(A$,";",1)
1930 IF K0=0 DO
1940 ' Y0=0:X0=0
1950 ELSE DO
1960 ' DEF FNF(X)=EVAL(MID$(A$,1,K0-1))
1970 ' X0=FNF(X)
1980 ' DEF FNF(X)=EVAL(IN$(A$,0,K0+1,""))
1990 ' Y0=FNF(X)
2000 ENDIF
2010 X0=ABS(X0):Y0=ABS(Y0)
2020 IF(X0>79)+(Y0>49)DO:ER=8
2030 ELSE MOVE X0,Y0
2040 ENDIF
2050 RETURN
2060 REM ********************************
2070 PROCEDURE "P"
2080 K0=LPOS(A$,";",1)
2090 IF K0=0 DO
2100 ' ER=8
2110 ELSE DO
2120 ' L0=VAL(MID$(A$,1,K0-1))
2130 ' K1=LPOS(A$,";",K0+1)
2140 ' IF K1=0 DO
2150 '   ER=8
2160 ' ELSE DO
2170 '   DEF FNF(X)=EVAL(MID$(A$,K0+1,K1-K0+1))
2180 '   X0=FNF(X)
2190 '   DEF FNF(X)=EVAL(IN$(A$,0,K0+K1+1,""))
2200 '   Y0=FNF(X)
2210 '   X0=ABS(X0):Y0=ABS(Y0)
2220 '   IF(X0>79)+(Y0>49)DO:ER=8
2230 '   ELSE PLOT L0,X0,Y0
2240 '   ENDIF
2250 ' ENDIF
2260 ENDIF
2270 RETURN
2280 REM ********************************
2290 FUNCTION "ESP"(D$)
2300 IF LEN(D$)>0 DO
2310 ' J0=LPOS(D$," ",1)
2320 ' WHILE J0>0 DO
2330 '   D$=IN$(D$,J0-1,J0+1,"")
2340 '   J0=LPOS(D$," ",1)
2350 ' WEND
2360 ENDIF
2370 RESULT=D$
2380 REM ********************************
2390 PROCEDURE "PP"
2400 WHILE(P$(PP)="")*(PP>0)DO
2410 ' PP=PP-1
2420 WEND
2430 IF(PP=0)*(P$(0)="")DO
2440 PP=-1
2450 ENDIF
2460 RETURN
2470 REM ********************************
2480 CLR:DIM P$(255),E$(10):MODE 1
2490 DIM L(20),L$(20),ST(20),U$(20)
2500 PP=-1:H$="0123456789":TEMPO 7
2510 U$="JTMAEUCVXHQDP"
2520 E$(1)="N° de ligne trop grand"
2530 E$(2)="Ligne sans :"
2540 E$(3)="Nom de fichier trop long"
2550 E$(4)="Instruction non valide"
2560 E$(5)="Etiquette inconnue"
2570 E$(6)="Pile de retour pleine"
2590 E$(8)="Valeur trop grande"
2600 E$(9)="Variable non definie"
2610 E$(10)="Erreur de syntaxe"
2620 PRINT"██Interpreteur PILOT"
2630 PRINT SIZE;" OCTETS":PRINT
2640 REPEAT
2650 ' REPEAT
2660 '   I0=0:ER=0
2670 '   IF NU>255 DO:F0=0
2680 '   ELSIF P$(NU)<>""DO:F0=0
2690 '   ENDIF
2700 '   IF F0=0 DO
2710 '     INPUT">";A$
2720 '   ELSE DO
2730 '     NU$=STR$(NU)
2740 '     PRINT">";NU$;:INPUT " ";A$
2750 '     IF A$="Z" DO
2760 '       F0=0
2770 '     ELSE DO
2780 '       A$=NU$+" "+A$:NU=NU+IC
2790 '     ENDIF
2800 '   ENDIF
2810 '   REPEAT
2820 '     I0=I0+1:C$=MID$(A$,I0,1)
2830 '   UNTIL LPOS(H$,C$,1)=0
2840 '   I0=I0-1
2850 '   IF I0>0 DO
2860 '     N0=VAL(LEFT$(A$,I0))
2870 '     A$=IN$(A$,0,I0+1,"")
2880 '     IF A$<>""DO
2890 '       IF N0<256 DO
2900 '         WHILE LEFT$(A$,1)=" " DO
2910 '           A$=IN$(A$,0,2,"")
2920 '         WEND
2930 '         IF LPOS(A$,":",1)>0 DO
2940 '           P$(N0)=A$:PP=PP SUP N0
2950 '         ELSE ER=2
2960 '         ENDIF
2970 '       ELSE ER=1
2980 '       ENDIF
2990 '     ELSE P$(N0)=""
3000 '       IF PP=N0 DO
3010 '         @"PP"
3020 '       ENDIF
3030 '     ENDIF
3040 '   ELSE C$=LEFT$(A$,1)
3050 '     CASE C$ OF
3060 '     WHEN"H"DO:@"HELP"
3070 '     WHEN"A"DO:@"AUTO"
3080 '     WHEN"R"DO:@"RUN"
3090 '     WHEN"N"DO:@"NEW"
3100 '     WHEN"E"DO:@"EXIT"
3110 '     WHEN"S"DO:@"SAVE"
3120 '     WHEN"G"DO:@"GET"
3130 '     WHEN"L"DO:@"LIST"
3140 '     WHEN"D"DO:@"DELETE"
3150 '     WHEN"P"DO:@"PRINTER"
3160 '     WHEN"M"DO:@"RENUM"
3170 '     WHEN"!"DO:PRINT"█"
3180 '     OTHERWISE
3190 '     FORX0=1TO10:MUSIC "⁻B0":NEXT
3200 '     ENDCASE
3210 '   ENDIF
3220 ' UNTIL ER<>0
3230 ' PRINT:PRINT E$(ER);" ligne ";I0
3240 ' PRINT P$(I0):PRINT
3250 UNTIL 0
3270 PROCEDURE "PRINTER"
3280 INPUT"Imprimante (O,N)?";R$
3290 PR=(R$="O")
3300 RETURN
3310 REM ********************************
3320 PROCEDURE "NEW"
3330 INPUT"Vraiment (O,N)?";R$
3340 IF R$="O" DO:NULL P$(0):PP=-1:ENDIF
3350 RETURN
3360 REM ********************************
3370 PROCEDURE "DELETE"
3380 INPUT"De la ligne ";N0
3390 INPUT" a la ligne ";M0
3400 N0=INT(ABS(N0)):M0=INT(ABS(M0))
3410 IF N0>M0 DO:EXC N0,M0:ENDIF
3420 M0=M0 INF 255
3430 FOR I0=N0 TO M0:P$(I0)="":NEXT
3440 PP=255:@"PP"
3450 RETURN
3460 REM ********************************
3470 PROCEDURE "LIST"
3480 INPUT"De la ligne ";N0
3490 INPUT" a la ligne ";M0
3500 N0=INT(ABS(N0)):M0=INT(ABS(M0))
3510 IF N0>M0 DO:EXC N0,M0:ENDIF
3520 M0=M0 INF 255 INF PP
3530 IF PP>0 DO
3540 ' FOR I0=N0 TO M0
3550 '   IF P$(I0)<>"" DO
3560 '     PRINT I0;" "+P$(I0)
3570 '     IF PR=-1 DO
3580 '       PRINT/P I0;" "+P$(I0)
3590 '     ENDIF
3600 '   ENDIF
3610 ' NEXT I0:PRINT
3620 ENDIF
3630 RETURN
3640 REM ********************************
3650 PROCEDURE "EXIT"
3660 PROCEDURE "Q"
3670 END
3680 REM ********************************
3690 PROCEDURE "HELP"
3700 PRINT"██LES COMMANDES SONT"
3710 PRINT"     H=HELP"
3720 PRINT"     R=RUN"
3730 PRINT"     N=NEW"
3740 PRINT"     E=EXIT"
3750 PRINT"     S=SAVE"
3760 PRINT"     G=GET"
3770 PRINT"     D=DELETE"
3780 PRINT"     A=AUTO"
3790 PRINT"     P=PRINTER"
3800 PRINT"     L=LIST"
3810 PRINT"     M=RENUM"
3820 PRINT
3830 RETURN
3840 REM ********************************
3850 PROCEDURE"SAVE"
3860 INPUT"TITRE:";N$
3870 IF(LEN(N$)<11)+(PP>-1)DO
3880 ' N$=N$+".PLT":WOPEN N$
3890 ' PRINT/T PP
3900 ' FOR I0=0 TO PP
3910 '   PRINT/T P$(I0)
3920 ' NEXT I0
3930 ' CLOSE
3940 ELSE ER=3
3970 REM ********************************
3980 PROCEDURE "GET"
3990 INPUT"TITRE:";N$
4000 IF (LEN(N$)<11)*(N$<>"-")DO
4010 ' N$=N$+".PLT":ROPEN N$
4020 ELSE ROPEN
4030 ENDIF
4040 INPUT/T PP:NULL P$(0)
4050 FOR I0=0 TO PP
4060 ' INPUT/T P$(I0)
4070 NEXT I0
4080 CLOSE
4090 RETURN
4100 REM ********************************
4110 PROCEDURE "RENUM"
4120 INPUT"Premiere ligne:";NU
4130 INPUT"Increment:";P0
4140 NU=INT(ABS(NU)):P0=INT(ABS(P0))
4150 IF(NU>255)+(P0=0)DO
4160 ' ER=10:RETURN
4170 ENDIF
4180 J0=0:DIM Q$(255)
4190 FOR I0=0 TO 255
4200 ' IF P$(I0)<>""DO
4210 '   Q$(J0)=P$(I0):J0=J0+1
4220 ' ENDIF
4230 NEXT I0
4240 IF J0>0 DO
4250 ' NULL P$(0):K0=NU:I0=0
4260 ' REPEAT
4270 '   P$(K0)=Q$(I0):K0=K0+P0:I0=I0+1
4280 ' UNTIL (I0=J0)+(K0>255)
4290 ENDIF
4300 CLR Q$(0):PP=255:@"PP"
4310 RETURN
4320 REM ********************************
4330 PROCEDURE "AUTO"
4340 INPUT"Premiere ligne:";NU:NU=INT(ABS(NU))
4350 IF NU<255 DO
4360 ' INPUT"et increment  :";IC
4370 ' IC=INT(ABS(IC)):F0=1
4380 ELSE ER=1:F0=0
4390 ENDIF
4400 RETURN
4410 REM ********************************
4420 PROCEDURE "RUN"
4430 IF PP=-1 DO:RETURN:ENDIF
4440 J0=0
4450 FOR I0=0 TO PP
4460 ' A$=P$(I0)
4470 ' IF A$<>""DO
4480 '   C$=LEFT$(A$,2)
4490 '   IF C$="*:"DO
4500 '     J0=J0+1:L(J0)=I0
4510 '     L$(J0)=IN$(A$,0,3,"")
4520 '   ENDIF
4530 ' ENDIF
4540 NEXT I0
4550 LT=J0:I0=-1:SP=0:V1=-1:NULL V$(0)
4560 REPEAT
4570 ' ER=0
4580 ' I0=I0+1:A$=P$(I0):C$=LEFT$(A$,1)
4590 ' M$=MID$(A$,2,1)
4600 ' TE=((M$="Y")*(MF=0))OR((M$="N")*(MF=1)
   )OR(A$="")OR(C$="*")OR(C$="R")
4610 ' IF TE=0 DO
4620 '   K0=LPOS(A$,":",1)
4630 '   A$=IN$(A$,0,K0+1,"")
4640 '   IF LPOS(U$,C$,1)=0 DO
4650 '     ER=4
4660 '   ELSE DO
4670 '     POKE 24444,ASC(C$):GOSUB40
4680 '   ENDIF
4690 ' ENDIF
4700 UNTIL(ER>0)+(I0=PP)
4710 RETURN
4720 REM ********************************
```



LE MONITOR SB 1510 TROISIEME ET DERNIERE

Cet article est le dernier de l'analyse du MONITOR du MZ80-B. Les fonctions qui restent à étudier sont celles relatives à l'horloge et à l'emission de sons. Nous terminerons cette étude en rappelant les « trucs » qui agissent sur le monitor, tous parus dans le SHARPENTIER. Vous aurez ainsi la possibilité de comprendre le fonctionnement. Si vous en trouvez d'autres écrivez-nous !

Nous commençons donc à l'adresse ODFB.

ODFB	TEMPO	correspond à l'ordre BASIC TEMPO. Charge TEMPW (001D) avec B — la valeur du tempo BASIC.
0E06	TIMST	mise à l'heure de l'horloge. AMPM (000F) est chargé avec 1 si t>12h, sinon avec 0. INICI (0016-0017) contient l'heure initiale exprimée en secondes.
0E51	TIMRD	blocage du compteur de temps, lecture puis ajout du temps initial (INICI).
0EBE	BELL	émission d'un BIP. Nombre de périodes 0030H, durée de la période 0060H (en cycles machine).
	appelle :	SOUT
0ECC	SOUT	émission d'un son caractérisé par les valeurs de BC (nombre de périodes = durée) et de HL (durée de la période = fréquence). Le haut-parleur est alimenté (en continu) par OUT (E3H), 05 puis non alimenté par OUT (E3H), 04.
	appelle :	SOUT2
0EE9	MELDY	émission de notes codées BASIC. (DE) est l'adresse du début de la chaîne à jouer, la fin est repérée par * ou CR (13H).
	appelle :	ONPU RYTHM
0F34	ONPU	recherche des caractéristiques de la note à émettre (durée et hauteur) en fonction de la lettre code, de l'octave, de éventuel et du tempo du morceau.
0F94	MTBL	table des hauteurs de notes
0FAC	M TBL	table des hauteurs des notes affectées de
0FDD	RYTHM	émission des notes décodées. (BC) : durée, (HL) : période.
	appelle :	SOUT

RAPPEL DES TRUCS MONITOR

```
POKE $0947,$05 l'écran ne s'efface plus par CONSOLE Snn,nn
POKE $0897,$00 SCROLL de l'écran ligne par ligne
POKE $06FC,$00 motorisation du curseur sans appui sur SHIFT
POKE $06DD,A   A<40H : le curseur clignote plus lentement
               A>40H : le curseur clignote plus rapidement
POKE $0725,A   A<40H : répétition du déplacement du curseur plus lente
               A>40H : répétition du déplacement du curseur plus rapide
POKE $097C,$00 plus de contrôle anti-rebond du clavier, il y a répétition automatique des caractères.
               Mais ATTENTION à la reprise en main ! Il faut remettre $A6 en $097C sinon gare au bégaiement !
POKE $0015,$00 émission de BIP à chaque touche enfoncée
POKE $0EC4,50 et POKE $0EC1,2 modification du BIP (plus discret...)
POKE $0EC4,A   modifie la hauteur du son émis par USR ($0EBE).
               A placer dans une boucle BASIC (FOR A=1 TO 50 par exemple)
               pour émettre des sons étranges.
POKE $0023,$2E et POKE $0024,$35 transforme le 00 du pavé numérique en .5 (utile pour entrer les notes par exemple.
USR($0517)     attente de 2 secondes
USR($0500)     attente de 4 secondes.
               Le délai peut être modifié en agissant sur les octets $0502 et $0503 (valeurs initiales 2B et 0F)
```

Nous terminerons le rappel de ces trucs en redonnant les quelques octets permettant d'afficher les flêches curseurs dans les chaînes (voir SHARPENTIER numéro 5).

```
0779   C3 90 13   JP 1390H
13BF   0D
1390   FE 07      CP 07
1392   30 07      JR NC,07H
1394   CB 5B      BIT 3,B
1396   28 03      JR Z,03
1398   C3 54 07   JP 0754H
1401   CD 6E 0A   CALL 0A6EH
1404   C3 62 07   JP 0762H
```

Ce programme est écrit pour le BASIC-DISK, et occupe la place du titre. Pour le BASIC SP-5510 il suffit de l'implanter à partir de $1350. De nombreuses autres modifications du monitor vous permettront de « personnaliser » votre MZ80B. En particulier il est facile de modifier la table des codes des touches de façon à avoir un clavier AZERTY. De même en modifiant des codes des hauteurs des notes, vous pouvez avoir un MZ jouant abominablement faux...

Jean MILLET

(NDLR) Dans le numéro 15, un superbe tableur de Jean MILLET, les autres, B'istes s'endormiraient-ils sur leur clavier ?

LA LOGITHEQUE

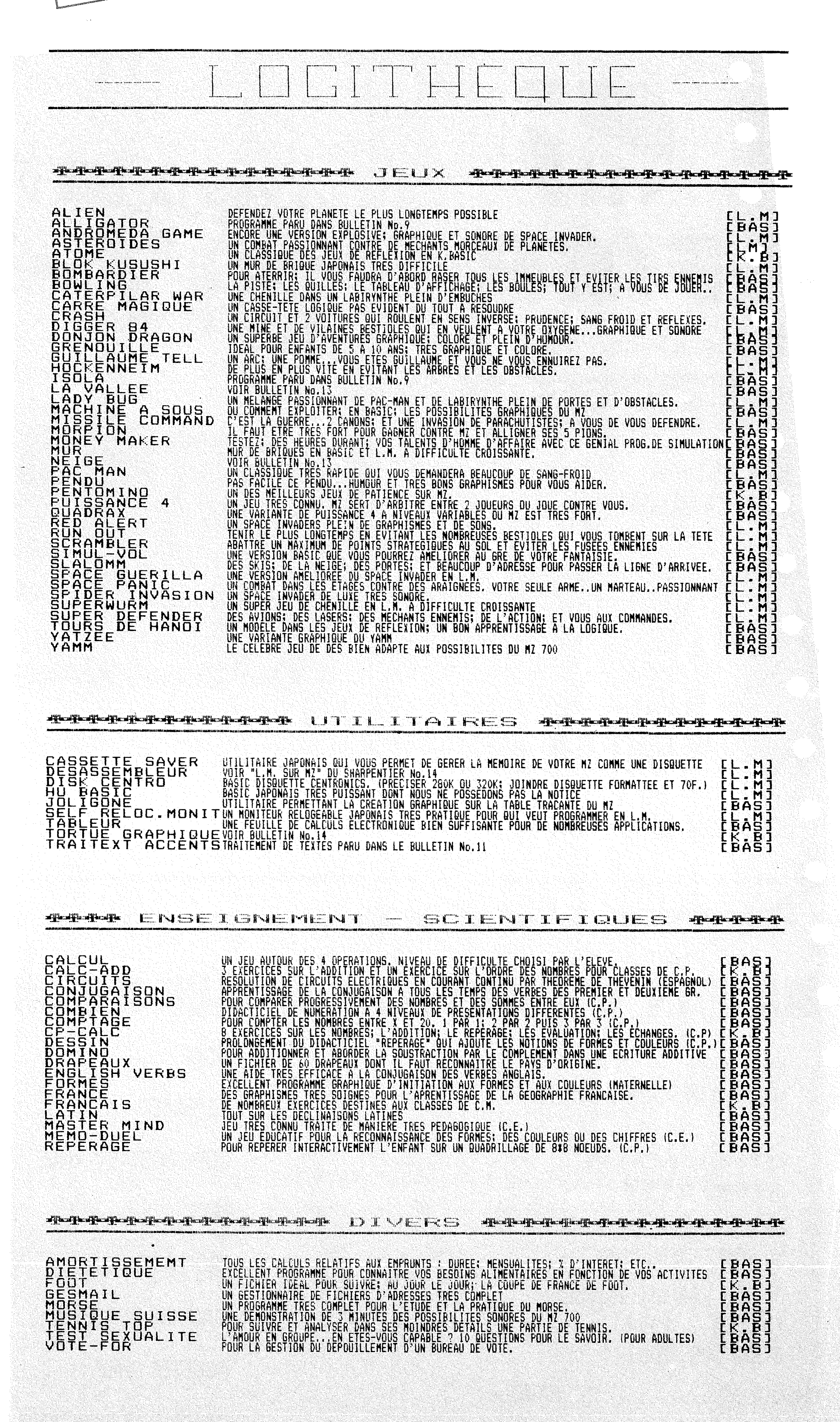
La logithèque est constituée en grande partie de logiciels conçus réalisés par les SHARPENTIERS et transmis au Club. Les modalités d'accès à celle-ci répondent à deux impératifs :

- **Assurer une légitime contrepartie aux concepteurs de programmes.**
- **En faire bénéficier tous les Sharpentiers, mêmes débutants en programmation.**

Dans cet esprit, nous avons créé la procédure suivante :

1. Vous nous faites parvenir un ou plusieurs programmes dont vous êtes l'auteur, nous vous retournons, sous 20 jours maximum un accusé de réception accompagné d'un nombre de points équivalent à la valeur de ce(s) programme(s), de (1 à 5 par programme), vous pourrez ensuite nous demander, en une seule fois, autant de programmes de la logithèque que de points acquis. Vos programmes doivent nous parvenir sur cassette, accompagnés d'une notice aussi complète que possible.

2. Vous désirez recevoir des programmes mais n'avez pas de points à votre crédit. Faites-nous parvenir, dans ce cas, la somme de 15 F par programme demandé, ainsi qu'une cassette vierge, nous vous retournerons vos programmes sous 20 jours maximum. OFFRE LIMITEE A 3 PROGRAMMES PAR ENVOI.



LOGITHEQUE

JEUX

ALIEN	DEFENDEZ VOTRE PLANETE LE PLUS LONGTEMPS POSSIBLE	[L.M]
ALLIGATOR	PROGRAMME PARU DANS BULLETIN No.9	[BAS]
ANDROMEDA GAME	ENCORE UNE VERSION EXPLOSIVE; GRAPHIQUE ET SONORE DE SPACE INVADER.	[L.M]
ASTEROIDES	UN COMBAT PASSIONNANT CONTRE DE MECHANTS MORCEAUX DE PLANETES.	[L.M]
ATOME	UN CLASSIQUE DES JEUX DE REFLEXION EN K.BASIC	[K.B]
BLOK KUSUSHI	UN MUR DE BRIQUE JAPONAIS TRES DIFFICILE	[L.M]
BOMBARDIER	POUR ATERRIR; IL VOUS FAUDRA D'ABORD RASER TOUS LES IMMEUBLES ET EVITER LES TIRS ENNEMIS	[BAS]
BOWLING	LA PISTE; LES QUILLES; LE TABLEAU D'AFFICHAGE; LES BOULES; TOUT Y EST; A VOUS DE JOUER..	[BAS]
CATERPILAR WAR	UNE CHENILLE DANS UN LABIRYNTHE PLEIN D'EMBUCHES	[L.M]
CARRE MAGIQUE	UN CASSE-TETE LOGIQUE PAS EVIDENT DU TOUT A RESOUDRE	[BAS]
CRASH	UN CIRCUIT ET 2 VOITURES QUI ROULENT EN SENS INVERSE; PRUDENCE; SANG FROID ET REFLEXES.	[L.M]
DIGGER 84	UNE MINE ET DE VILAINES BESTIOLES QUI EN VEULENT A VOTRE OXYGENE...GRAPHIQUE ET SONORE	[L.M]
DONJON DRAGON	UN SUPERBE JEU D'AVENTURES GRAPHIQUE; COLORE ET PLEIN D'HUMOUR.	[BAS]
GRENOUILLE	IDEAL POUR ENFANTS DE 5 A 10 ANS; TRES GRAPHIQUE ET COLORE.	[BAS]
GUILLAUME TELL	UN ARC; UNE POMME...VOUS ETES GUILLAUME ET VOUS NE VOUS ENNUIREZ PAS.	[L.M]
HOCKENNEIM	DE PLUS EN PLUS VITE EN EVITANT LES ARBRES ET LES OBSTACLES.	[L.M]
ISOLA	PROGRAMME PARU DANS BULLETIN No.9	[BAS]
LA VALLEE	VOIR BULLETIN No.13	[BAS]
LADY BUG	UN MELANGE PASSIONNANT DE PAC-MAN ET DE LABIRYNTHE PLEIN DE PORTES ET D'OBSTACLES.	[L.M]
MACHINE A SOUS	OU COMMENT EXPLOITER; EN BASIC; LES POSSIBILITES GRAPHIQUES DU MZ	[BAS]
MISSILE COMMAND	C'EST LA GUERRE...2 CANONS; ET UNE INVASION DE PARACHUTISTES; A VOUS DE VOUS DEFENDRE.	[L.M]
MORPION	IL FAUT ETRE TRES FORT POUR GAGNER CONTRE MZ ET ALLIGNER SES 5 PIONS.	[BAS]
MONEY MAKER	TESTEZ; DES HEURES DURANT; VOS TALENTS D'HOMME D'AFFAIRE AVEC CE GENIAL PROG.DE SIMULATION	[BAS]
MUR	MUR DE BRIQUES EN BASIC ET L.M. A DIFFICULTE CROISSANTE.	[BAS]
NEIGE	VOIR BULLETIN No.13	[BAS]
PAC MAN	UN CLASSIQUE TRES RAPIDE QUI VOUS DEMANDERA BEAUCOUP DE SANG-FROID	[L.M]
PENDU	PAS FACILE CE PENDU...HUMOUR ET TRES BONS GRAPHISMES POUR VOUS AIDER.	[BAS]
PENTOMINO	UN DES MEILLEURS JEUX DE PATIENCE SUR MZ.	[K.B]
PUISSANCE 4	UN JEU TRES CONNU. MZ SERT D'ARBITRE ENTRE 2 JOUEURS OU JOUE CONTRE VOUS.	[BAS]
QUADRAX	UNE VARIANTE DE PUISSANCE 4 A NIVEAUX VARIABLES OU MZ EST TRES FORT.	[BAS]
RED ALERT	UN SPACE INVADERS PLEIN DE GRAPHISMES ET DE SONS.	[L.M]
RUN OUT	TENIR LE PLUS LONGTEMPS EN EVITANT LES NOMBREUSES BESTIOLES QUI VOUS TOMBENT SUR LA TETE	[L.M]
SCRAMBLER	ABATTRE UN MAXIMUM DE POINTS STRATEGIQUES AU SOL ET EVITER LES FUSEES ENNEMIES	[L.M]
SIMUL-VOL	UNE VERSION BASIC QUE VOUS POURREZ AMELIORER AU GRE DE VOTRE FANTAISIE.	[BAS]
SLALOMM	DES SKIS; DE LA NEIGE; DES PORTES; ET BEAUCOUP D'ADRESSE POUR PASSER LA LIGNE D'ARRIVEE.	[BAS]
SPACE GUERILLA	UNE VERSION AMELIOREE DU SPACE INVADER EN L.M.	[L.M]
SPACE PANIC	UN COMBAT DANS LES ETAGES CONTRE DES ARAIGNEES. VOTRE SEULE ARME..UN MARTEAU..PASSIONNANT	[L.M]
SPIDER INVASION	UN SPACE INVADER DE LUXE TRES SONORE.	[L.M]
SUPERWURM	UN SUPER JEU DE CHENILLE EN L.M. A DIFFICULTE CROISSANTE	[L.M]
SUPER DEFENDER	DES AVIONS; DES LASERS; DES MECHANTS ENNEMIS; DE L'ACTION; ET VOUS AUX COMMANDES.	[L.M]
TOURS DE HANOI	UN MODELE DANS LES JEUX DE REFLEXION; UN BON APPRENTISSAGE A LA LOGIQUE.	[BAS]
YATZEE	UNE VARIANTE GRAPHIQUE DU YAMM	[BAS]
YAMM	LE CELEBRE JEU DE DES BIEN ADAPTE AUX POSSIBILITES DU MZ 700	[BAS]

UTILITAIRES

CASSETTE SAVER	UTILITAIRE JAPONAIS QUI VOUS PERMET DE GERER LA MEMOIRE DE VOTRE MZ COMME UNE DISQUETTE	[L.M]
DESASSEMBLEUR	VOIR "L.M. SUR MZ" DU SHARPENTIER No.14	[L.M]
DISK CENTRO	BASIC DISQUETTE CENTRONICS. (PRECISER 280K OU 320K; JOINDRE DISQUETTE FORMATTEE ET 70F.)	[L.M]
HU BASIC	BASIC JAPONAIS TRES PUISSANT DONT NOUS NE POSSEDONS PAS LA NOTICE	[L.M]
JOLIGONE	UTILITAIRE PERMETTANT LA CREATION GRAPHIQUE SUR LA TABLE TRACANTE DU MZ	[BAS]
SELF RELOC.MONIT	UN MONITEUR RELOGEABLE JAPONAIS TRES PRATIQUE POUR QUI VEUT PROGRAMMER EN L.M.	[L.M]
TABLEUR	UNE FEUILLE DE CALCULS ELECTRONIQUE BIEN SUFFISANTE POUR DE NOMBREUSES APPLICATIONS.	[BAS]
TORTUE GRAPHIQUE	VOIR BULLETIN No.14	[K.B]
TRAITEXT ACCENTS	TRAITEMENT DE TEXTES PARU DANS LE BULLETIN No.11	[BAS]

ENSEIGNEMENT — SCIENTIFIQUES

CALCUL	UN JEU AUTOUR DES 4 OPERATIONS. NIVEAU DE DIFFICULTE CHOISI PAR L'ELEVE.	[BAS]
CALC-ADD	3 EXERCICES SUR L'ADDITION ET UN EXERCICE SUR L'ORDRE DES NOMBRES POUR CLASSES DE C.P.	[K.B]
CIRCUITS	RESOLUTION DE CIRCUITS ELECTRIQUES EN COURANT CONTINU PAR THEOREME DE THEVENIN (ESPAGNOL)	[BAS]
CONJUGAISON	APPRENTISSAGE DE LA CONJUGAISON A TOUS LES TEMPS DES VERBES DES PREMIER ET DEUXIEME GR.	[BAS]
COMPARAISONS	POUR COMPARER PROGRESSIVEMENT DES NOMBRES ET DES SOMMES ENTRE EUX (C.P.)	[BAS]
COMBIEN	DIDACTICIEL DE NUMERATION A 4 NIVEAUX DE PRESENTATIONS DIFFERENTES (C.P.)	[BAS]
COMPTAGE	POUR COMPTER LES NOMBRES ENTRE X ET 20. 1 PAR 1; 2 PAR 2 PUIS 3 PAR 3 (C.P.)	[BAS]
CP-CALC	8 EXERCICES SUR LES NOMBRES; L'ADDITION; LE REPERAGE; LES EVALUATION; LES ECHANGES. (C.P)	[K.B]
DESSIN	PROLONGEMENT DU DIDACTICIEL "REPERAGE" QUI AJOUTE LES NOTIONS DE FORMES ET COULEURS (C.P.)	[BAS]
DOMINO	POUR ADDITIONNER ET ABORDER LA SOUSTRACTION PAR LE COMPLEMENT DANS UNE ECRITURE ADDITIVE	[BAS]
DRAPEAUX	UN FICHIER DE 60 DRAPEAUX DONT IL FAUT RECONNAITRE LE PAYS D'ORIGINE.	[BAS]
ENGLISH VERBS	UNE AIDE TRES EFFICACE A LA CONJUGAISON DES VERBES ANGLAIS.	[BAS]
FORMES	EXCELLENT PROGRAMME GRAPHIQUE D'INITIATION AUX FORMES ET AUX COULEURS (MATERNELLE)	[BAS]
FRANCE	DES GRAPHISMES TRES SOIGNES POUR L'APRENTISSAGE DE LA GEOGRAPHIE FRANCAISE.	[BAS]
FRANCAIS	DE NOMBREUX EXERCICES DESTINES AUX CLASSES DE C.M.	[K.B]
LATIN	TOUT SUR LES DECLINAISONS LATINES	[BAS]
MASTER MIND	JEU TRES CONNU TRAITE DE MANIERE TRES PEDAGOGIQUE (C.E.)	[BAS]
MEMO-DUEL	UN JEU EDUCATIF POUR LA RECONNAISSANCE DES FORMES; DES COULEURS OU DES CHIFFRES (C.E.)	[BAS]
REPERAGE	POUR REPERER INTERACTIVEMENT L'ENFANT SUR UN QUADRILLAGE DE 8*8 NOEUDS. (C.P.)	[BAS]

DIVERS

AMORTISSEMEMT	TOUS LES CALCULS RELATIFS AUX EMPRUNTS : DUREE; MENSUALITES; % D'INTERET; ETC..	[BAS]
DIETETIQUE	EXCELLENT PROGRAMME POUR CONNAITRE VOS BESOINS ALIMENTAIRES EN FONCTION DE VOS ACTIVITES	[BAS]
FOOT	UN FICHIER IDEAL POUR SUIVRE; AU JOUR LE JOUR; LA COUPE DE FRANCE DE FOOT.	[K.B]
GESMAIL	UN GESTIONNAIRE DE FICHIERS D'ADRESSES TRES COMPLET	[BAS]
MORSE	UN PROGRAMME TRES COMPLET POUR L'ETUDE ET LA PRATIQUE DU MORSE.	[BAS]
MUSIQUE SUISSE	UNE DEMONSTRATION DE 3 MINUTES DES POSSIBILITES SONORES DU MZ 700	[BAS]
TENNIS TOP	POUR SUIVRE ET ANALYSER DANS SES MOINDRES DETAILS UNE PARTIE DE TENNIS.	[K.B]
TEST SEXUALITE	L'AMOUR EN GROUPE...EN ETES-VOUS CAPABLE ? 10 QUESTIONS POUR LE SAVOIR. (POUR ADULTES)	[BAS]
VOTE-FOR	POUR LA GESTION DU DEPOUILLEMENT D'UN BUREAU DE VOTE.	[BAS]

BULLETIN D'INSCRIPTION AU CLUB DES SHARPENTIERS

☐ Je m'inscris
au CLUB DES SHARPENTIERS

Je bénéficie de tous les
avantages du CLUB

Je suis abonné pour 1 AN
au BULLETIN du CLUB

Je vous joins mon règlement
☐ FRANCE : 160 F
☐ ETRANGER : 200 F

CHEQUE N° .... BANQUE ......

DATE SIGNATURE

n° 14

NOM .... PRÉNOM ....

ADRESSE ....

CODE POSTAL / VILLE ....

PAYS ....

PROFESSION .... ÂGE ....

MACHINE POSSEDÉE .... DEPUIS ....

ACHETÉE CHEZ ....

UTILISATION PRINCIPALE DE VOTRE MACHINE ....

....

Club des Sharpentiers 151/153, avenue Jean-Jaurès 93307 AUBERVILLIERS CEDEX Tél : 834.93.44